

general\_ledger

Generated by Doxygen 1.8.1.2

Sun Jun 8 2014 13:28:27



# Contents

<b>1</b>	<b>General Ledger.</b>	<b>1</b>
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Data Structure Documentation</b>	<b>9</b>
4.1	ds_list Struct Reference . . . . .	9
4.1.1	Detailed Description . . . . .	9
4.1.2	Field Documentation . . . . .	9
4.1.2.1	current . . . . .	10
4.1.2.2	data_destructor . . . . .	10
4.1.2.3	free_on_delete . . . . .	10
4.1.2.4	head . . . . .	10
4.1.2.5	length . . . . .	10
4.1.2.6	tail . . . . .	10
4.2	ds_list_element Struct Reference . . . . .	10
4.2.1	Detailed Description . . . . .	10
4.2.2	Field Documentation . . . . .	11
4.2.2.1	data . . . . .	11
4.2.2.2	next . . . . .	11
4.2.2.3	previous . . . . .	11
4.3	ds_map Struct Reference . . . . .	11
4.3.1	Detailed Description . . . . .	11
4.3.2	Field Documentation . . . . .	12
4.3.2.1	hash_size . . . . .	12
4.3.2.2	lists . . . . .	12
4.4	ds_map_str Struct Reference . . . . .	12
4.4.1	Detailed Description . . . . .	12
4.4.2	Field Documentation . . . . .	12

4.4.2.1	hash_size	13
4.4.2.2	lists	13
4.5	ds_record Struct Reference	13
4.5.1	Detailed Description	13
4.5.2	Field Documentation	13
4.5.2.1	fields	13
4.6	ds_recordset Struct Reference	14
4.6.1	Detailed Description	14
4.6.2	Field Documentation	14
4.6.2.1	field_lengths	14
4.6.2.2	headers	14
4.6.2.3	num_fields	14
4.6.2.4	records	15
4.6.2.5	types	15
4.7	ds_str Struct Reference	15
4.7.1	Detailed Description	15
4.7.2	Field Documentation	15
4.7.2.1	capacity	15
4.7.2.2	data	15
4.7.2.3	length	15
4.8	ds_vector Struct Reference	15
4.8.1	Detailed Description	16
4.8.2	Field Documentation	16
4.8.2.1	current	16
4.8.2.2	data	16
4.8.2.3	data_destructor	16
4.8.2.4	free_on_delete	16
4.8.2.5	size	16
4.9	kv_pair_node Struct Reference	16
4.9.1	Detailed Description	17
4.9.2	Field Documentation	17
4.9.2.1	key	17
4.9.2.2	key	17
4.9.2.3	next	17
4.9.2.4	value	17
4.9.2.5	value	17
4.10	params Struct Reference	17
4.10.1	Detailed Description	18
4.10.2	Field Documentation	18
4.10.2.1	database	18

4.10.2.2	hostname	18
4.10.2.3	password	18
4.10.2.4	username	18
<b>5</b>	<b>File Documentation</b>	<b>19</b>
5.1	config.c File Reference	19
5.1.1	Detailed Description	20
5.1.2	Macro Definition Documentation	20
5.1.2.1	_XOPEN_SOURCE	20
5.1.3	Function Documentation	20
5.1.3.1	get_cmdline_options	20
5.1.3.2	get_configuration	20
5.1.3.3	params_free	21
5.1.3.4	params_init	21
5.2	config.h File Reference	21
5.2.1	Detailed Description	22
5.2.2	Function Documentation	22
5.2.2.1	get_cmdline_options	22
5.2.2.2	get_configuration	23
5.2.2.3	params_free	23
5.2.2.4	params_init	23
5.3	lib/database/database.h File Reference	23
5.3.1	Detailed Description	24
5.4	lib/database/db_connection.h File Reference	24
5.4.1	Detailed Description	25
5.4.2	Function Documentation	25
5.4.2.1	db_connect	25
5.5	lib/database/db_entities.c File Reference	25
5.5.1	Detailed Description	26
5.5.2	Function Documentation	26
5.5.2.1	db_create_entities_table	26
5.5.2.2	db_drop_entities_table	26
5.5.2.3	db_list_entities_report	27
5.6	lib/database/db_entities.h File Reference	27
5.6.1	Detailed Description	28
5.6.2	Function Documentation	28
5.6.2.1	db_create_entities_table	28
5.6.2.2	db_drop_entities_table	28
5.6.2.3	db_list_entities_report	28
5.7	lib/database/db_internal.h File Reference	28

5.7.1	Detailed Description	29
5.8	lib/database/db_jelines.c File Reference	29
5.8.1	Detailed Description	30
5.8.2	Function Documentation	30
5.8.2.1	db_create_jelines_table	30
5.8.2.2	db_drop_jelines_table	30
5.8.2.3	db_list_jelines_report	31
5.9	lib/database/db_jelines.h File Reference	31
5.9.1	Detailed Description	32
5.9.2	Function Documentation	32
5.9.2.1	db_create_jelines_table	32
5.9.2.2	db_drop_jelines_table	32
5.9.2.3	db_list_jelines_report	32
5.10	lib/database/db_jes.c File Reference	32
5.10.1	Detailed Description	33
5.10.2	Function Documentation	33
5.10.2.1	db_create_jes_table	33
5.10.2.2	db_drop_jes_table	33
5.10.2.3	db_list_jes_report	34
5.11	lib/database/db_jes.h File Reference	34
5.11.1	Detailed Description	35
5.11.2	Function Documentation	35
5.11.2.1	db_create_jes_table	35
5.11.2.2	db_drop_jes_table	35
5.11.2.3	db_list_jes_report	35
5.12	lib/database/db_jesrcs.c File Reference	35
5.12.1	Detailed Description	36
5.12.2	Function Documentation	36
5.12.2.1	db_create_jesrcs_table	36
5.12.2.2	db_drop_jesrcs_table	36
5.12.2.3	db_list_jesrcs_report	37
5.13	lib/database/db_jesrcs.h File Reference	37
5.13.1	Detailed Description	38
5.13.2	Function Documentation	38
5.13.2.1	db_create_jesrcs_table	38
5.13.2.2	db_drop_jesrcs_table	38
5.13.2.3	db_list_jesrcs_report	38
5.14	lib/database/db_nomaccts.c File Reference	38
5.14.1	Detailed Description	39
5.14.2	Function Documentation	39

5.14.2.1	<a href="#">db_create_nomaccts_table</a>	39
5.14.2.2	<a href="#">db_drop_nomaccts_table</a>	39
5.14.2.3	<a href="#">db_list_nomaccts_report</a>	40
5.15	<a href="#">lib/database/db_nomaccts.h File Reference</a>	40
5.15.1	<a href="#">Detailed Description</a>	41
5.15.2	<a href="#">Function Documentation</a>	41
5.15.2.1	<a href="#">db_create_nomaccts_table</a>	41
5.15.2.2	<a href="#">db_drop_nomaccts_table</a>	41
5.15.2.3	<a href="#">db_list_nomaccts_report</a>	41
5.16	<a href="#">lib/database/db_query.h File Reference</a>	42
5.16.1	<a href="#">Detailed Description</a>	42
5.16.2	<a href="#">Function Documentation</a>	43
5.16.2.1	<a href="#">db_execute_query</a>	43
5.17	<a href="#">lib/database/db_reporting.c File Reference</a>	43
5.17.1	<a href="#">Detailed Description</a>	43
5.17.2	<a href="#">Function Documentation</a>	44
5.17.2.1	<a href="#">db_create_report_from_query</a>	44
5.17.2.2	<a href="#">db_current_trial_balance_report</a>	44
5.18	<a href="#">lib/database/db_reporting.h File Reference</a>	44
5.18.1	<a href="#">Detailed Description</a>	44
5.18.2	<a href="#">Function Documentation</a>	45
5.18.2.1	<a href="#">db_create_recordset_from_query</a>	45
5.18.2.2	<a href="#">db_create_report_from_query</a>	45
5.18.2.3	<a href="#">db_current_trial_balance_report</a>	45
5.19	<a href="#">lib/database/db_sampledata.c File Reference</a>	45
5.19.1	<a href="#">Detailed Description</a>	46
5.20	<a href="#">lib/database/db_sampledata.h File Reference</a>	46
5.20.1	<a href="#">Detailed Description</a>	47
5.21	<a href="#">lib/database/db_sql.h File Reference</a>	47
5.21.1	<a href="#">Detailed Description</a>	48
5.21.2	<a href="#">Function Documentation</a>	49
5.21.2.1	<a href="#">db_create_entities_table_sql</a>	49
5.21.2.2	<a href="#">db_create_jelines_table_sql</a>	49
5.21.2.3	<a href="#">db_create_jes_table_sql</a>	49
5.21.2.4	<a href="#">db_create_jesrcs_table_sql</a>	49
5.21.2.5	<a href="#">db_create_nomaccts_table_sql</a>	49
5.21.2.6	<a href="#">db_create_standingdata_table_sql</a>	49
5.21.2.7	<a href="#">db_create_users_table_sql</a>	50
5.21.2.8	<a href="#">db_current_trial_balance_report_sql</a>	50
5.21.2.9	<a href="#">db_drop_entities_table_sql</a>	50

5.21.2.10 db_drop_jelines_table_sql . . . . .	50
5.21.2.11 db_drop_jes_table_sql . . . . .	50
5.21.2.12 db_drop_jesrcs_table_sql . . . . .	50
5.21.2.13 db_drop_nomaccts_table_sql . . . . .	50
5.21.2.14 db_drop_standingdata_table_sql . . . . .	51
5.21.2.15 db_drop_users_table_sql . . . . .	51
5.21.2.16 db_list_entities_report_sql . . . . .	51
5.21.2.17 db_list_jelines_report_sql . . . . .	51
5.21.2.18 db_list_jes_report_sql . . . . .	51
5.21.2.19 db_list_jesrcs_report_sql . . . . .	51
5.21.2.20 db_list_nomaccts_report_sql . . . . .	51
5.21.2.21 db_list_users_report_sql . . . . .	52
5.21.2.22 db_show_standingdata_report_sql . . . . .	52
5.22 lib/database/db_standingdata.c File Reference . . . . .	52
5.22.1 Detailed Description . . . . .	52
5.22.2 Function Documentation . . . . .	53
5.22.2.1 db_create_standingdata_table . . . . .	53
5.22.2.2 db_drop_standingdata_table . . . . .	53
5.22.2.3 db_show_standingdata_report . . . . .	53
5.23 lib/database/db_standingdata.h File Reference . . . . .	54
5.23.1 Detailed Description . . . . .	54
5.23.2 Function Documentation . . . . .	55
5.23.2.1 db_create_standingdata_table . . . . .	55
5.23.2.2 db_drop_standingdata_table . . . . .	55
5.23.2.3 db_show_standingdata_report . . . . .	55
5.24 lib/database/db_structure.c File Reference . . . . .	55
5.24.1 Detailed Description . . . . .	56
5.24.2 Function Documentation . . . . .	56
5.24.2.1 db_create_database_structure . . . . .	56
5.24.2.2 db_delete_database_structure . . . . .	56
5.25 lib/database/db_structure.h File Reference . . . . .	57
5.25.1 Detailed Description . . . . .	57
5.25.2 Function Documentation . . . . .	58
5.25.2.1 db_create_database_structure . . . . .	58
5.25.2.2 db_delete_database_structure . . . . .	58
5.26 lib/database/db_users.c File Reference . . . . .	58
5.26.1 Detailed Description . . . . .	58
5.26.2 Function Documentation . . . . .	59
5.26.2.1 db_create_users_table . . . . .	59
5.26.2.2 db_drop_users_table . . . . .	59



5.26.2.3 db_list_users_report . . . . .	59
5.27 lib/database/db_users.h File Reference . . . . .	60
5.27.1 Detailed Description . . . . .	60
5.27.2 Function Documentation . . . . .	61
5.27.2.1 db_create_users_table . . . . .	61
5.27.2.2 db_drop_users_table . . . . .	61
5.27.2.3 db_list_users_report . . . . .	61
5.28 lib/database/dummy/db_dummy_create_entities_table_sql.c File Reference . . . . .	61
5.28.1 Detailed Description . . . . .	61
5.28.2 Function Documentation . . . . .	62
5.28.2.1 db_create_entities_table_sql . . . . .	62
5.29 lib/database/dummy/db_dummy_create_users_table_sql.c File Reference . . . . .	62
5.29.1 Detailed Description . . . . .	62
5.29.2 Function Documentation . . . . .	62
5.29.2.1 db_create_users_table_sql . . . . .	62
5.30 lib/database/dummy/db_dummy_drop_entities_table_sql.c File Reference . . . . .	62
5.30.1 Detailed Description . . . . .	63
5.30.2 Function Documentation . . . . .	63
5.30.2.1 db_drop_entities_table_sql . . . . .	63
5.31 lib/database/dummy/db_dummy_drop_users_table_sql.c File Reference . . . . .	63
5.31.1 Detailed Description . . . . .	63
5.31.2 Function Documentation . . . . .	63
5.31.2.1 db_drop_users_table_sql . . . . .	63
5.32 lib/database/dummy/db_dummy_general.c File Reference . . . . .	64
5.32.1 Detailed Description . . . . .	64
5.32.2 Macro Definition Documentation . . . . .	65
5.32.2.1 _XOPEN_SOURCE . . . . .	65
5.32.3 Function Documentation . . . . .	65
5.32.3.1 db_connect . . . . .	65
5.32.3.2 db_create_recordset_from_query . . . . .	65
5.32.3.3 db_execute_query . . . . .	65
5.33 lib/database/dummy/db_dummy_list_entities_report_sql.c File Reference . . . . .	65
5.33.1 Detailed Description . . . . .	66
5.33.2 Function Documentation . . . . .	66
5.33.2.1 db_list_entities_report_sql . . . . .	66
5.34 lib/database/dummy/db_dummy_list_users_report_sql.c File Reference . . . . .	66
5.34.1 Detailed Description . . . . .	66
5.34.2 Function Documentation . . . . .	67
5.34.2.1 db_list_users_report_sql . . . . .	67
5.35 lib/database/mysql/db_mysql_create_entities_table_sql.c File Reference . . . . .	67

5.35.1 Detailed Description . . . . .	67
5.35.2 Function Documentation . . . . .	67
5.35.2.1 db_create_entities_table_sql . . . . .	67
5.36 lib/database/mysql/db_mysql_create_jelines_table_sql.c File Reference . . . . .	67
5.36.1 Detailed Description . . . . .	68
5.36.2 Function Documentation . . . . .	68
5.36.2.1 db_create_jelines_table_sql . . . . .	68
5.37 lib/database/mysql/db_mysql_create_jes_table_sql.c File Reference . . . . .	68
5.37.1 Detailed Description . . . . .	68
5.37.2 Function Documentation . . . . .	68
5.37.2.1 db_create_jes_table_sql . . . . .	68
5.38 lib/database/mysql/db_mysql_create_jesrcs_table_sql.c File Reference . . . . .	69
5.38.1 Detailed Description . . . . .	69
5.38.2 Function Documentation . . . . .	69
5.38.2.1 db_create_jesrcs_table_sql . . . . .	69
5.39 lib/database/mysql/db_mysql_create_nomaccts_table_sql.c File Reference . . . . .	69
5.39.1 Detailed Description . . . . .	69
5.39.2 Function Documentation . . . . .	70
5.39.2.1 db_create_nomaccts_table_sql . . . . .	70
5.40 lib/database/mysql/db_mysql_create_standingdata_table_sql.c File Reference . . . . .	70
5.40.1 Detailed Description . . . . .	70
5.40.2 Function Documentation . . . . .	70
5.40.2.1 db_create_standingdata_table_sql . . . . .	70
5.41 lib/database/mysql/db_mysql_create_users_table_sql.c File Reference . . . . .	70
5.41.1 Detailed Description . . . . .	71
5.41.2 Function Documentation . . . . .	71
5.41.2.1 db_create_users_table_sql . . . . .	71
5.42 lib/database/mysql/db_mysql_current_trial_balance_report_sql.c File Reference . . . . .	71
5.42.1 Detailed Description . . . . .	71
5.42.2 Function Documentation . . . . .	71
5.42.2.1 db_current_trial_balance_report_sql . . . . .	71
5.43 lib/database/mysql/db_mysql_drop_entities_table_sql.c File Reference . . . . .	72
5.43.1 Detailed Description . . . . .	72
5.43.2 Function Documentation . . . . .	72
5.43.2.1 db_drop_entities_table_sql . . . . .	72
5.44 lib/database/mysql/db_mysql_drop_jelines_table_sql.c File Reference . . . . .	72
5.44.1 Detailed Description . . . . .	72
5.44.2 Function Documentation . . . . .	73
5.44.2.1 db_drop_jelines_table_sql . . . . .	73
5.45 lib/database/mysql/db_mysql_drop_jes_table_sql.c File Reference . . . . .	73

5.45.1 Detailed Description . . . . .	73
5.45.2 Function Documentation . . . . .	73
5.45.2.1 db_drop_jes_table_sql . . . . .	73
5.46 lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c File Reference . . . . .	73
5.46.1 Detailed Description . . . . .	74
5.46.2 Function Documentation . . . . .	74
5.46.2.1 db_drop_jesrcs_table_sql . . . . .	74
5.47 lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c File Reference . . . . .	74
5.47.1 Detailed Description . . . . .	74
5.47.2 Function Documentation . . . . .	74
5.47.2.1 db_drop_nomaccts_table_sql . . . . .	74
5.48 lib/database/mysql/db_mysql_drop_standingdata_table_sql.c File Reference . . . . .	75
5.48.1 Detailed Description . . . . .	75
5.48.2 Function Documentation . . . . .	75
5.48.2.1 db_drop_standingdata_table_sql . . . . .	75
5.49 lib/database/mysql/db_mysql_drop_users_table_sql.c File Reference . . . . .	75
5.49.1 Detailed Description . . . . .	75
5.49.2 Function Documentation . . . . .	76
5.49.2.1 db_drop_users_table_sql . . . . .	76
5.50 lib/database/mysql/db_mysql_general.c File Reference . . . . .	76
5.50.1 Detailed Description . . . . .	77
5.50.2 Function Documentation . . . . .	77
5.50.2.1 db_connect . . . . .	77
5.50.2.2 db_create_recordset_from_query . . . . .	77
5.50.2.3 db_execute_query . . . . .	77
5.50.3 Variable Documentation . . . . .	78
5.50.3.1 conn_mss . . . . .	78
5.50.3.2 main_mss . . . . .	78
5.51 lib/database/mysql/db_mysql_list_entities_report_sql.c File Reference . . . . .	78
5.51.1 Detailed Description . . . . .	78
5.51.2 Function Documentation . . . . .	78
5.51.2.1 db_list_entities_report_sql . . . . .	78
5.52 lib/database/mysql/db_mysql_list_jelines_report_sql.c File Reference . . . . .	78
5.52.1 Detailed Description . . . . .	79
5.52.2 Function Documentation . . . . .	79
5.52.2.1 db_list_jelines_report_sql . . . . .	79
5.53 lib/database/mysql/db_mysql_list_jes_report_sql.c File Reference . . . . .	79
5.53.1 Detailed Description . . . . .	79
5.53.2 Function Documentation . . . . .	79
5.53.2.1 db_list_jes_report_sql . . . . .	79

5.54	<a href="#">lib/database/mysql/db_mysql_list_jesrcs_report_sql.c File Reference</a>	80
5.54.1	Detailed Description	80
5.54.2	Function Documentation	80
5.54.2.1	<a href="#">db_list_jesrcs_report_sql</a>	80
5.55	<a href="#">lib/database/mysql/db_mysql_list_nomaccts_report_sql.c File Reference</a>	80
5.55.1	Detailed Description	80
5.55.2	Function Documentation	81
5.55.2.1	<a href="#">db_list_nomaccts_report_sql</a>	81
5.56	<a href="#">lib/database/mysql/db_mysql_list_users_report_sql.c File Reference</a>	81
5.56.1	Detailed Description	81
5.56.2	Function Documentation	81
5.56.2.1	<a href="#">db_list_users_report_sql</a>	81
5.57	<a href="#">lib/database/mysql/db_mysql_show_standingdata_report_sql.c File Reference</a>	81
5.57.1	Detailed Description	82
5.57.2	Function Documentation	82
5.57.2.1	<a href="#">db_show_standingdata_report_sql</a>	82
5.58	<a href="#">lib/datastruct/data_structures.h File Reference</a>	82
5.58.1	Detailed Description	83
5.59	<a href="#">lib/datastruct/ds_fieldtypes.h File Reference</a>	83
5.59.1	Detailed Description	83
5.59.2	Enumeration Type Documentation	84
5.59.2.1	<a href="#">ds_field_types</a>	84
5.60	<a href="#">lib/datastruct/ds_list.c File Reference</a>	84
5.60.1	Detailed Description	85
5.60.2	Function Documentation	85
5.60.2.1	<a href="#">ds_list_append</a>	85
5.60.2.2	<a href="#">ds_list_create</a>	85
5.60.2.3	<a href="#">ds_list_destroy</a>	86
5.60.2.4	<a href="#">ds_list_destructor</a>	86
5.60.2.5	<a href="#">ds_list_element</a>	86
5.60.2.6	<a href="#">ds_list_get_next_data</a>	86
5.60.2.7	<a href="#">ds_list_get_prev_data</a>	87
5.60.2.8	<a href="#">ds_list_is_empty</a>	87
5.60.2.9	<a href="#">ds_list_length</a>	87
5.60.2.10	<a href="#">ds_list_remove_all</a>	87
5.60.2.11	<a href="#">ds_list_remove_tail</a>	88
5.60.2.12	<a href="#">ds_list_seek_end</a>	88
5.60.2.13	<a href="#">ds_list_seek_start</a>	88
5.61	<a href="#">lib/datastruct/ds_list.h File Reference</a>	88
5.61.1	Detailed Description	89

5.61.2	Typedef Documentation . . . . .	89
5.61.2.1	ds_list . . . . .	89
5.61.3	Function Documentation . . . . .	90
5.61.3.1	ds_list_append . . . . .	90
5.61.3.2	ds_list_create . . . . .	90
5.61.3.3	ds_list_destroy . . . . .	90
5.61.3.4	ds_list_destructor . . . . .	90
5.61.3.5	ds_list_element . . . . .	91
5.61.3.6	ds_list_get_next_data . . . . .	91
5.61.3.7	ds_list_get_prev_data . . . . .	91
5.61.3.8	ds_list_is_empty . . . . .	91
5.61.3.9	ds_list_length . . . . .	92
5.61.3.10	ds_list_remove_all . . . . .	92
5.61.3.11	ds_list_remove_tail . . . . .	92
5.61.3.12	ds_list_seek_end . . . . .	92
5.61.3.13	ds_list_seek_start . . . . .	92
5.62	lib/datastruct/ds_map.c File Reference . . . . .	92
5.62.1	Detailed Description . . . . .	94
5.62.2	Function Documentation . . . . .	94
5.62.2.1	ds_map_destroy . . . . .	94
5.62.2.2	ds_map_get_value . . . . .	94
5.62.2.3	ds_map_init . . . . .	94
5.62.2.4	ds_map_insert . . . . .	94
5.62.2.5	ds_map_print_all . . . . .	95
5.63	lib/datastruct/ds_map.h File Reference . . . . .	95
5.63.1	Detailed Description . . . . .	96
5.63.2	Typedef Documentation . . . . .	96
5.63.2.1	ds_map . . . . .	96
5.63.3	Function Documentation . . . . .	96
5.63.3.1	ds_map_destroy . . . . .	96
5.63.3.2	ds_map_get_value . . . . .	96
5.63.3.3	ds_map_init . . . . .	97
5.63.3.4	ds_map_insert . . . . .	97
5.63.3.5	ds_map_print_all . . . . .	97
5.64	lib/datastruct/ds_map_str.c File Reference . . . . .	97
5.64.1	Detailed Description . . . . .	98
5.64.2	Function Documentation . . . . .	99
5.64.2.1	ds_map_str_destroy . . . . .	99
5.64.2.2	ds_map_str_get_value . . . . .	99
5.64.2.3	ds_map_str_init . . . . .	99

5.64.2.4	<a href="#">ds_map_str_insert</a>	99
5.65	<a href="#">lib/datastruct/ds_map_str.h File Reference</a>	99
5.65.1	<a href="#">Detailed Description</a>	100
5.65.2	<a href="#">Typedef Documentation</a>	101
5.65.2.1	<a href="#">ds_map_str</a>	101
5.65.3	<a href="#">Function Documentation</a>	101
5.65.3.1	<a href="#">ds_map_str_destroy</a>	101
5.65.3.2	<a href="#">ds_map_str_get_value</a>	101
5.65.3.3	<a href="#">ds_map_str_init</a>	101
5.65.3.4	<a href="#">ds_map_str_insert</a>	101
5.66	<a href="#">lib/datastruct/ds_record.c File Reference</a>	102
5.66.1	<a href="#">Detailed Description</a>	103
5.66.2	<a href="#">Function Documentation</a>	103
5.66.2.1	<a href="#">ds_record_clear</a>	103
5.66.2.2	<a href="#">ds_record_create</a>	103
5.66.2.3	<a href="#">ds_record_destroy</a>	103
5.66.2.4	<a href="#">ds_record_destructor</a>	104
5.66.2.5	<a href="#">ds_record_get_field</a>	104
5.66.2.6	<a href="#">ds_record_get_next_data</a>	104
5.66.2.7	<a href="#">ds_record_make_delim_string</a>	104
5.66.2.8	<a href="#">ds_record_make_values_string</a>	105
5.66.2.9	<a href="#">ds_record_seek_start</a>	105
5.66.2.10	<a href="#">ds_record_set_field</a>	105
5.66.2.11	<a href="#">ds_record_size</a>	105
5.66.2.12	<a href="#">ds_record_tokenize</a>	105
5.67	<a href="#">lib/datastruct/ds_record.h File Reference</a>	106
5.67.1	<a href="#">Detailed Description</a>	107
5.67.2	<a href="#">Typedef Documentation</a>	107
5.67.2.1	<a href="#">ds_record</a>	107
5.67.3	<a href="#">Function Documentation</a>	107
5.67.3.1	<a href="#">ds_record_clear</a>	107
5.67.3.2	<a href="#">ds_record_create</a>	107
5.67.3.3	<a href="#">ds_record_destroy</a>	108
5.67.3.4	<a href="#">ds_record_destructor</a>	108
5.67.3.5	<a href="#">ds_record_get_field</a>	108
5.67.3.6	<a href="#">ds_record_get_next_data</a>	108
5.67.3.7	<a href="#">ds_record_make_delim_string</a>	109
5.67.3.8	<a href="#">ds_record_make_values_string</a>	109
5.67.3.9	<a href="#">ds_record_seek_start</a>	109
5.67.3.10	<a href="#">ds_record_set_field</a>	109

5.67.3.11 ds_record_size . . . . .	109
5.67.3.12 ds_record_tokenize . . . . .	110
5.68 lib/datastruct/ds_recordset.c File Reference . . . . .	110
5.68.1 Detailed Description . . . . .	111
5.68.2 Function Documentation . . . . .	111
5.68.2.1 ds_recordset_add_record . . . . .	111
5.68.2.2 ds_recordset_create . . . . .	112
5.68.2.3 ds_recordset_destroy . . . . .	112
5.68.2.4 ds_recordset_get_next_insert_query . . . . .	112
5.68.2.5 ds_recordset_get_text_report . . . . .	112
5.68.2.6 ds_recordset_next_record . . . . .	112
5.68.2.7 ds_recordset_num_fields . . . . .	113
5.68.2.8 ds_recordset_num_records . . . . .	113
5.68.2.9 ds_recordset_seek_start . . . . .	113
5.68.2.10 ds_recordset_set_headers . . . . .	113
5.68.2.11 ds_recordset_set_type . . . . .	113
5.69 lib/datastruct/ds_recordset.h File Reference . . . . .	114
5.69.1 Detailed Description . . . . .	115
5.69.2 Typedef Documentation . . . . .	115
5.69.2.1 ds_recordset . . . . .	115
5.69.3 Function Documentation . . . . .	115
5.69.3.1 ds_recordset_add_record . . . . .	115
5.69.3.2 ds_recordset_create . . . . .	116
5.69.3.3 ds_recordset_destroy . . . . .	116
5.69.3.4 ds_recordset_get_next_insert_query . . . . .	116
5.69.3.5 ds_recordset_get_text_report . . . . .	116
5.69.3.6 ds_recordset_next_record . . . . .	116
5.69.3.7 ds_recordset_num_fields . . . . .	117
5.69.3.8 ds_recordset_num_records . . . . .	117
5.69.3.9 ds_recordset_seek_start . . . . .	117
5.69.3.10 ds_recordset_set_headers . . . . .	117
5.69.3.11 ds_recordset_set_type . . . . .	118
5.70 lib/datastruct/ds_str.c File Reference . . . . .	118
5.70.1 Detailed Description . . . . .	120
5.70.2 Function Documentation . . . . .	120
5.70.2.1 ds_str_assign . . . . .	120
5.70.2.2 ds_str_assign_cstr . . . . .	120
5.70.2.3 ds_str_char_at_index . . . . .	120
5.70.2.4 ds_str_clear . . . . .	121
5.70.2.5 ds_str_compare . . . . .	121

5.70.2.6	<a href="#">ds_str_compare_cstr</a>	121
5.70.2.7	<a href="#">ds_str_concat</a>	121
5.70.2.8	<a href="#">ds_str_concat_cstr</a>	122
5.70.2.9	<a href="#">ds_str_create</a>	122
5.70.2.10	<a href="#">ds_str_create_direct</a>	122
5.70.2.11	<a href="#">ds_str_create_sprintf</a>	122
5.70.2.12	<a href="#">ds_str_cstr</a>	123
5.70.2.13	<a href="#">ds_str_decorate</a>	123
5.70.2.14	<a href="#">ds_str_destroy</a>	123
5.70.2.15	<a href="#">ds_str_destructor</a>	123
5.70.2.16	<a href="#">ds_str_doubleval</a>	124
5.70.2.17	<a href="#">ds_str_dup</a>	124
5.70.2.18	<a href="#">ds_str_getline</a>	124
5.70.2.19	<a href="#">ds_str_hash</a>	124
5.70.2.20	<a href="#">ds_str_intval</a>	125
5.70.2.21	<a href="#">ds_str_is_alnum</a>	125
5.70.2.22	<a href="#">ds_str_is_empty</a>	125
5.70.2.23	<a href="#">ds_str_length</a>	125
5.70.2.24	<a href="#">ds_str_size_to_fit</a>	126
5.70.2.25	<a href="#">ds_str_split</a>	126
5.70.2.26	<a href="#">ds_str_strchr</a>	126
5.70.2.27	<a href="#">ds_str_substr_left</a>	126
5.70.2.28	<a href="#">ds_str_substr_right</a>	127
5.70.2.29	<a href="#">ds_str_trim</a>	127
5.70.2.30	<a href="#">ds_str_trim_leading</a>	127
5.70.2.31	<a href="#">ds_str_trim_trailing</a>	127
5.70.2.32	<a href="#">ds_str_trunc</a>	127
5.71	<a href="#">lib/datastruct/ds_str.h File Reference</a>	128
5.71.1	<a href="#">Detailed Description</a>	130
5.71.2	<a href="#">Typedef Documentation</a>	130
5.71.2.1	<a href="#">ds_str</a>	130
5.71.3	<a href="#">Function Documentation</a>	130
5.71.3.1	<a href="#">ds_str_assign</a>	130
5.71.3.2	<a href="#">ds_str_assign_cstr</a>	130
5.71.3.3	<a href="#">ds_str_char_at_index</a>	130
5.71.3.4	<a href="#">ds_str_clear</a>	131
5.71.3.5	<a href="#">ds_str_compare</a>	131
5.71.3.6	<a href="#">ds_str_compare_cstr</a>	131
5.71.3.7	<a href="#">ds_str_concat</a>	131
5.71.3.8	<a href="#">ds_str_concat_cstr</a>	132



5.71.3.9	<a href="#">ds_str_create</a>	132
5.71.3.10	<a href="#">ds_str_create_direct</a>	132
5.71.3.11	<a href="#">ds_str_create_sprintf</a>	132
5.71.3.12	<a href="#">ds_str_cstr</a>	133
5.71.3.13	<a href="#">ds_str_decorate</a>	133
5.71.3.14	<a href="#">ds_str_destroy</a>	133
5.71.3.15	<a href="#">ds_str_destructor</a>	133
5.71.3.16	<a href="#">ds_str_doubleval</a>	133
5.71.3.17	<a href="#">ds_str_dup</a>	134
5.71.3.18	<a href="#">ds_str_getline</a>	134
5.71.3.19	<a href="#">ds_str_hash</a>	134
5.71.3.20	<a href="#">ds_str_intval</a>	134
5.71.3.21	<a href="#">ds_str_is_alnum</a>	135
5.71.3.22	<a href="#">ds_str_is_empty</a>	135
5.71.3.23	<a href="#">ds_str_length</a>	135
5.71.3.24	<a href="#">ds_str_size_to_fit</a>	135
5.71.3.25	<a href="#">ds_str_split</a>	136
5.71.3.26	<a href="#">ds_str_strchr</a>	136
5.71.3.27	<a href="#">ds_str_substr_left</a>	136
5.71.3.28	<a href="#">ds_str_substr_right</a>	136
5.71.3.29	<a href="#">ds_str_trim</a>	137
5.71.3.30	<a href="#">ds_str_trim_leading</a>	137
5.71.3.31	<a href="#">ds_str_trim_trailing</a>	137
5.71.3.32	<a href="#">ds_str_trunc</a>	137
5.72	<a href="#">lib/datastruct/ds_vector.c File Reference</a>	137
5.72.1	<a href="#">Detailed Description</a>	138
5.72.2	<a href="#">Function Documentation</a>	139
5.72.2.1	<a href="#">ds_vector_clear</a>	139
5.72.2.2	<a href="#">ds_vector_create</a>	139
5.72.2.3	<a href="#">ds_vector_destroy</a>	139
5.72.2.4	<a href="#">ds_vector_destructor</a>	139
5.72.2.5	<a href="#">ds_vector_element</a>	140
5.72.2.6	<a href="#">ds_vector_get_next_data</a>	140
5.72.2.7	<a href="#">ds_vector_seek_start</a>	140
5.72.2.8	<a href="#">ds_vector_set</a>	140
5.72.2.9	<a href="#">ds_vector_size</a>	141
5.73	<a href="#">lib/datastruct/ds_vector.h File Reference</a>	141
5.73.1	<a href="#">Detailed Description</a>	142
5.73.2	<a href="#">Typedef Documentation</a>	142
5.73.2.1	<a href="#">ds_vector</a>	142

5.73.3	Function Documentation	142
5.73.3.1	ds_vector_clear	142
5.73.3.2	ds_vector_create	142
5.73.3.3	ds_vector_destroy	143
5.73.3.4	ds_vector_destructor	143
5.73.3.5	ds_vector_element	143
5.73.3.6	ds_vector_get_next_data	143
5.73.3.7	ds_vector_seek_start	144
5.73.3.8	ds_vector_set	144
5.73.3.9	ds_vector_size	144
5.74	lib/file_ops/config_file_read.c File Reference	144
5.74.1	Detailed Description	145
5.74.2	Macro Definition Documentation	146
5.74.2.1	CONFIG_MAP_SIZE	146
5.74.2.2	MAX_BUFFER_SIZE	146
5.74.3	Function Documentation	146
5.74.3.1	config_file_read	146
5.74.3.2	config_free	146
5.74.3.3	config_init	146
5.74.3.4	config_value_get	146
5.74.3.5	config_value_get_cstr	147
5.74.3.6	config_value_set	147
5.75	lib/file_ops/config_file_read.h File Reference	147
5.75.1	Detailed Description	148
5.75.2	Macro Definition Documentation	149
5.75.2.1	CONFIG_FILE_MALFORMED_FILE	149
5.75.2.2	CONFIG_FILE_NO_FILE	149
5.75.2.3	CONFIG_FILE_OK	149
5.75.3	Function Documentation	149
5.75.3.1	config_file_read	149
5.75.3.2	config_free	149
5.75.3.3	config_init	149
5.75.3.4	config_value_get	149
5.75.3.5	config_value_get_cstr	150
5.75.3.6	config_value_set	150
5.76	lib/file_ops/delim_file_read.c File Reference	150
5.76.1	Detailed Description	151
5.76.2	Macro Definition Documentation	151
5.76.2.1	MAX_LINE_SIZE	151
5.76.3	Function Documentation	152

5.76.3.1	<a href="#">delim_file_read</a>	152
5.77	<a href="#">lib/file_ops/delim_file_read.h File Reference</a>	152
5.77.1	Detailed Description	153
5.77.2	Function Documentation	153
5.77.2.1	<a href="#">delim_file_read</a>	153
5.78	<a href="#">lib/file_ops/file_ops.h File Reference</a>	154
5.78.1	Detailed Description	154
5.79	<a href="#">lib/gl_general/gl_errors.c File Reference</a>	155
5.79.1	Detailed Description	155
5.79.2	Function Documentation	156
5.79.2.1	<a href="#">gl_error_quit</a>	156
5.80	<a href="#">lib/gl_general/gl_errors.h File Reference</a>	156
5.80.1	Detailed Description	156
5.80.2	Function Documentation	156
5.80.2.1	<a href="#">gl_error_quit</a>	156
5.81	<a href="#">lib/gl_general/gl_general.h File Reference</a>	156
5.81.1	Detailed Description	157
5.82	<a href="#">lib/gl_general/gl_logging.c File Reference</a>	157
5.82.1	Detailed Description	158
5.82.2	Function Documentation	158
5.82.2.1	<a href="#">gl_log_msg</a>	158
5.82.2.2	<a href="#">gl_set_logging</a>	158
5.83	<a href="#">lib/gl_general/gl_logging.h File Reference</a>	159
5.83.1	Detailed Description	159
5.83.2	Function Documentation	160
5.83.2.1	<a href="#">gl_log_msg</a>	160
5.83.2.2	<a href="#">gl_set_logging</a>	160
5.84	<a href="#">main.c File Reference</a>	160
5.84.1	Detailed Description	161
5.84.2	Function Documentation	161
5.84.2.1	<a href="#">login</a>	161
5.84.2.2	<a href="#">main</a>	161
5.84.2.3	<a href="#">print_help_message</a>	161
5.84.2.4	<a href="#">print_usage_message</a>	162
5.84.2.5	<a href="#">print_version_message</a>	162
5.84.2.6	<a href="#">test_functionality</a>	162



## Chapter 1

# General Ledger.

General Ledger will be a fully-featured, multi-user, open-source general ledger system. The project is in the early stages of development.



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">ds_list</a>	9
<a href="#">ds_list_element</a>	10
<a href="#">ds_map</a>	11
<a href="#">ds_map_str</a>	12
<a href="#">ds_record</a>	13
<a href="#">ds_recordset</a>	14
<a href="#">ds_str</a>	15
<a href="#">ds_vector</a>	15
<a href="#">kv_pair_node</a>	16
<a href="#">params</a>	17





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">config.c</a>	Implementation of program configuration functionality . . . . .	19
<a href="#">config.h</a>	Interface to program configuration functionality . . . . .	21
<a href="#">main.c</a>	Main function for general_ledger . . . . .	160
<a href="#">lib/database/database.h</a>	User interface to database functionality . . . . .	23
<a href="#">lib/database/db_connection.h</a>	Interface to database connection functionality . . . . .	24
<a href="#">lib/database/db_entities.c</a>	Implementation of entities functionality . . . . .	25
<a href="#">lib/database/db_entities.h</a>	Interface to entities functionality . . . . .	27
<a href="#">lib/database/db_internal.h</a>	Internal library interface to database functionality . . . . .	28
<a href="#">lib/database/db_jelines.c</a>	Implementation of journal entries functionality . . . . .	29
<a href="#">lib/database/db_jelines.h</a>	Interface to journal entry lines functionality . . . . .	31
<a href="#">lib/database/db_jes.c</a>	Implementation of journal entries functionality . . . . .	32
<a href="#">lib/database/db_jes.h</a>	Interface to journal entries functionality . . . . .	34
<a href="#">lib/database/db_jesrcs.c</a>	Implementation of journal entry sources functionality . . . . .	35
<a href="#">lib/database/db_jesrcs.h</a>	Interface to journal entry sources functionality . . . . .	37
<a href="#">lib/database/db_nomaccts.c</a>	Implementation of nominal accounts functionality . . . . .	38
<a href="#">lib/database/db_nomaccts.h</a>	Interface to nominal accounts functionality . . . . .	40
<a href="#">lib/database/db_query.h</a>	Interface to database query functionality . . . . .	42
<a href="#">lib/database/db_reporting.c</a>	Implementation of database reporting functionality . . . . .	43
<a href="#">lib/database/db_reporting.h</a>	Interface to database reporting functionality . . . . .	44

lib/database/db_sampledata.c	
Implementation of database sample data functionality . . . . .	45
lib/database/db_sampledata.h	
Interface to database sample data functionality . . . . .	46
lib/database/db_sql.h	
Interface to database specific SQL strings . . . . .	47
lib/database/db_standingdata.c	
Implementation of standing data functionality . . . . .	52
lib/database/db_standingdata.h	
Interface to journal entries functionality . . . . .	54
lib/database/db_structure.c	
Implementation of database structure functionality . . . . .	55
lib/database/db_structure.h	
Interface to database structure functionality . . . . .	57
lib/database/db_users.c	
Implementation of users functionality . . . . .	58
lib/database/db_users.h	
Interface to users functionality . . . . .	60
lib/database/dummy/db_dummy_create_entities_table_sql.c	
Returns dummy SQL query to create entities table . . . . .	61
lib/database/dummy/db_dummy_create_users_table_sql.c	
Returns dummy SQL query to create users table . . . . .	62
lib/database/dummy/db_dummy_drop_entities_table_sql.c	
Returns dummy SQL query to drop entities table . . . . .	62
lib/database/dummy/db_dummy_drop_users_table_sql.c	
Returns dummy SQL query to drop users table . . . . .	63
lib/database/dummy/db_dummy_general.c	
Implementation of dummy database functionality . . . . .	64
lib/database/dummy/db_dummy_list_entities_report_sql.c	
Returns dummy SQL query to create list entities report . . . . .	65
lib/database/dummy/db_dummy_list_users_report_sql.c	
Returns dummy SQL query to create list users report . . . . .	66
lib/database/mysql/db_mysql_create_entities_table_sql.c	
Returns MYSQL SQL query to create entities table . . . . .	67
lib/database/mysql/db_mysql_create_jelines_table_sql.c	
Returns MYSQL SQL query to create journal entry lines table . . . . .	67
lib/database/mysql/db_mysql_create_jes_table_sql.c	
Returns MYSQL SQL query to create journal entries table . . . . .	68
lib/database/mysql/db_mysql_create_jesrcs_table_sql.c	
Returns MYSQL SQL query to create JE sources table . . . . .	69
lib/database/mysql/db_mysql_create_nomaccts_table_sql.c	
Returns MYSQL SQL query to create nominal accounts table . . . . .	69
lib/database/mysql/db_mysql_create_standingdata_table_sql.c	
Returns MYSQL SQL query to create standing data table . . . . .	70
lib/database/mysql/db_mysql_create_users_table_sql.c	
Returns MYSQL SQL query to create users table . . . . .	70
lib/database/mysql/db_mysql_current_trial_balance_report_sql.c	
Returns MYSQL SQL query to create current TB report . . . . .	71
lib/database/mysql/db_mysql_drop_entities_table_sql.c	
Returns MYSQL SQL query to drop entities table . . . . .	72
lib/database/mysql/db_mysql_drop_jelines_table_sql.c	
Returns MYSQL SQL query to drop journal entry lines table . . . . .	72
lib/database/mysql/db_mysql_drop_jes_table_sql.c	
Returns MYSQL SQL query to drop entities table . . . . .	73
lib/database/mysql/db_mysql_drop_jesrcs_table_sql.c	
Returns MYSQL SQL query to drop JE sources table . . . . .	73
lib/database/mysql/db_mysql_drop_nomaccts_table_sql.c	
Returns MYSQL SQL query to drop nominal accounts table . . . . .	74

lib/database/mysql/db_mysql_drop_standingdata_table_sql.c	
Returns MYSQL SQL query to drop standing data table . . . . .	75
lib/database/mysql/db_mysql_drop_users_table_sql.c	
Returns MYSQL SQL query to drop users table . . . . .	75
lib/database/mysql/db_mysql_general.c	
Implementation of MYSQL database functionality . . . . .	76
lib/database/mysql/db_mysql_list_entities_report_sql.c	
Returns MYSQL SQL query to create list entities report . . . . .	78
lib/database/mysql/db_mysql_list_jelines_report_sql.c	
Returns MYSQL SQL query to create JE lines report . . . . .	78
lib/database/mysql/db_mysql_list_jes_report_sql.c	
Returns MYSQL SQL query to create journal entries report . . . . .	79
lib/database/mysql/db_mysql_list_jesrcs_report_sql.c	
Returns MYSQL SQL query to create JE sources report . . . . .	80
lib/database/mysql/db_mysql_list_nomaccts_report_sql.c	
Returns MYSQL SQL query to create list nominal accounts report . . . . .	80
lib/database/mysql/db_mysql_list_users_report_sql.c	
Returns MYSQL SQL query to create list users report . . . . .	81
lib/database/mysql/db_mysql_show_standingdata_report_sql.c	
Returns MYSQL SQL query to create show standing data report . . . . .	81
lib/datastruct/data_structures.h	
Interface to data structures . . . . .	82
lib/datastruct/ds_fieldtypes.h	
Record field types enumeration . . . . .	83
lib/datastruct/ds_list.c	
Implementation of generic doubly-linked list data structure . . . . .	84
lib/datastruct/ds_list.h	
Interface to generic doubly-linked list data structure . . . . .	88
lib/datastruct/ds_map.c	
Implementation of string-string hash map data structure . . . . .	92
lib/datastruct/ds_map.h	
Interface to string-string hash map data structure . . . . .	95
lib/datastruct/ds_map_str.c	
Implementation of string-string hash map data structure . . . . .	97
lib/datastruct/ds_map_str.h	
Interface to string-string hash map data structure . . . . .	99
lib/datastruct/ds_record.c	
Implementation of record database structure . . . . .	102
lib/datastruct/ds_record.h	
Interface to record data structure . . . . .	106
lib/datastruct/ds_recordset.c	
Implementation of query result set structure . . . . .	110
lib/datastruct/ds_recordset.h	
Interface to record set structure . . . . .	114
lib/datastruct/ds_str.c	
Implementation of string data structure . . . . .	118
lib/datastruct/ds_str.h	
Interface to string data structure . . . . .	128
lib/datastruct/ds_vector.c	
Implementation of generic doubly-linked vector data structure . . . . .	137
lib/datastruct/ds_vector.h	
Interface to generic doubly-linked vector data structure . . . . .	141
lib/file_ops/config_file_read.c	
Implementation of configuration file reading functionality . . . . .	144
lib/file_ops/config_file_read.h	
Interface to configuration file reading functionality . . . . .	147
lib/file_ops/delim_file_read.c	
Implementation of delimited file reading functionality . . . . .	150

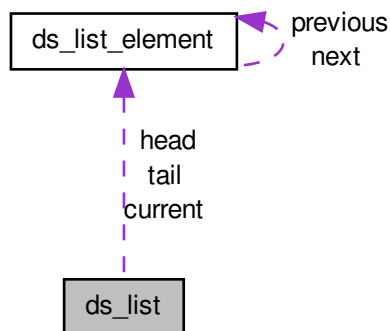
lib/file_ops/ <a href="#">delim_file_read.h</a>	
Interface to delimited file reading functionality . . . . .	152
lib/file_ops/ <a href="#">file_ops.h</a>	
User interface to file operations functionality . . . . .	154
lib/gl_general/ <a href="#">gl_errors.c</a>	
Implementation of error functionality . . . . .	155
lib/gl_general/ <a href="#">gl_errors.h</a>	
Interface to error functionality . . . . .	156
lib/gl_general/ <a href="#">gl_general.h</a>	
User interface to logging and error functionality . . . . .	156
lib/gl_general/ <a href="#">gl_logging.c</a>	
Implementation of logging functionality . . . . .	157
lib/gl_general/ <a href="#">gl_logging.h</a>	
Interface to logging functionality . . . . .	159

## Chapter 4

# Data Structure Documentation

### 4.1 ds\_list Struct Reference

Collaboration diagram for ds\_list:



#### Data Fields

- `size_t` `length`
- `bool` `free_on_delete`
- `struct ds_list_element *` `head`
- `struct ds_list_element *` `tail`
- `struct ds_list_element *` `current`
- `void(* data_destructor )(void *)`

#### 4.1.1 Detailed Description

List data structure

#### 4.1.2 Field Documentation

#### 4.1.2.1 struct `ds_list_element*` `ds_list::current`

Pointer to current element

#### 4.1.2.2 void(\* `ds_list::data_destructor`)(void \*)

Data destructor function

#### 4.1.2.3 bool `ds_list::free_on_delete`

'Free on delete' flag

#### 4.1.2.4 struct `ds_list_element*` `ds_list::head`

Pointer to head element

#### 4.1.2.5 size\_t `ds_list::length`

Length of list

#### 4.1.2.6 struct `ds_list_element*` `ds_list::tail`

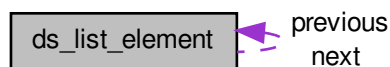
Pointer to tail element

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_list.c](#)

## 4.2 ds\_list\_element Struct Reference

Collaboration diagram for `ds_list_element`:



### Data Fields

- void \* [data](#)
- struct [ds\\_list\\_element](#) \* [previous](#)
- struct [ds\\_list\\_element](#) \* [next](#)

#### 4.2.1 Detailed Description

List element data structure

## 4.2.2 Field Documentation

### 4.2.2.1 void\* ds\_list\_element::data

Pointer to data

### 4.2.2.2 struct ds\_list\_element\* ds\_list\_element::next

Pointer to next element

### 4.2.2.3 struct ds\_list\_element\* ds\_list\_element::previous

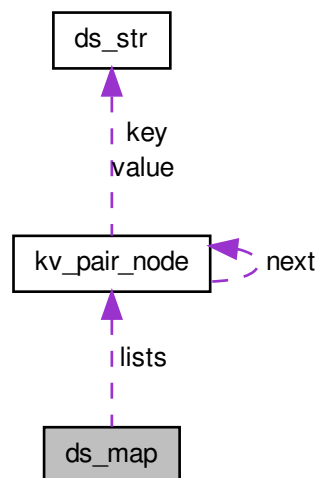
Pointer to previous element

The documentation for this struct was generated from the following file:

- lib/datastruct/[ds\\_list.c](#)

## 4.3 ds\_map Struct Reference

Collaboration diagram for ds\_map:



### Data Fields

- struct [kv\\_pair\\_node](#) \*\* `lists`
- size\_t [hash\\_size](#)

### 4.3.1 Detailed Description

Structure to hold a hash map

### 4.3.2 Field Documentation

#### 4.3.2.1 `size_t ds_map::hash_size`

Size of array of lists

#### 4.3.2.2 `struct kv_pair_node** ds_map::lists`

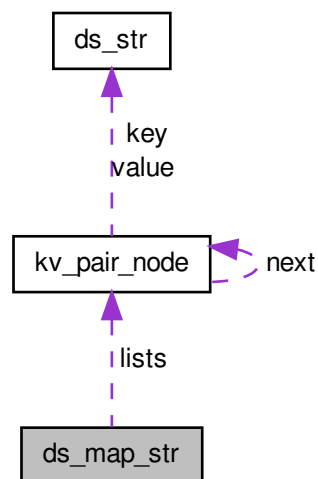
Pointer to array of lists

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_map.c](#)

## 4.4 `ds_map_str` Struct Reference

Collaboration diagram for `ds_map_str`:



### Data Fields

- struct [kv\\_pair\\_node](#) \*\* `lists`
- `size_t` [hash\\_size](#)

#### 4.4.1 Detailed Description

Structure to hold a hash map

#### 4.4.2 Field Documentation



#### 4.4.2.1 `size_t ds_map_str::hash_size`

Size of array of lists

#### 4.4.2.2 `struct kv_pair_node** ds_map_str::lists`

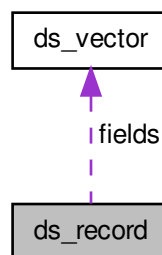
Pointer to array of lists

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_map\\_str.c](#)

## 4.5 ds\_record Struct Reference

Collaboration diagram for ds\_record:



### Data Fields

- struct [ds\\_vector](#) \* `fields`

#### 4.5.1 Detailed Description

Vector data structure

#### 4.5.2 Field Documentation

##### 4.5.2.1 `struct ds_vector* ds_record::fields`

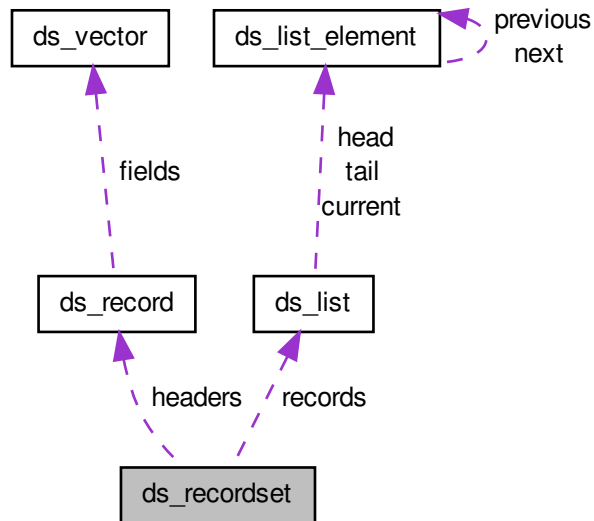
Vector of fields

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_record.c](#)

## 4.6 ds\_recordset Struct Reference

Collaboration diagram for ds\_recordset:



### Data Fields

- `size_t num_fields`
- `size_t * field_lengths`
- `ds_record headers`
- `ds_list records`
- `enum ds_field_types * types`

#### 4.6.1 Detailed Description

Result set structure

#### 4.6.2 Field Documentation

##### 4.6.2.1 `size_t* ds_recordset::field_lengths`

Lengths of the longest fields

##### 4.6.2.2 `ds_record ds_recordset::headers`

A list of field headers

##### 4.6.2.3 `size_t ds_recordset::num_fields`

The number of fields in a record

#### 4.6.2.4 ds\_list ds\_recordset::records

A list of records

#### 4.6.2.5 enum ds\_field\_types\* ds\_recordset::types

Types of records

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_recordset.c](#)

## 4.7 ds\_str Struct Reference

### Data Fields

- `char *` [data](#)
- `size_t` [length](#)
- `size_t` [capacity](#)

#### 4.7.1 Detailed Description

Structure to contain string

#### 4.7.2 Field Documentation

##### 4.7.2.1 size\_t ds\_str::capacity

The size of the `data` buffer

##### 4.7.2.2 char\* ds\_str::data

The data in C-style string format

##### 4.7.2.3 size\_t ds\_str::length

The length of the string

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_str.c](#)

## 4.8 ds\_vector Struct Reference

### Data Fields

- `size_t` [size](#)
- `size_t` [current](#)
- `bool` [free\\_on\\_delete](#)
- `void **` [data](#)
- `void(*` [data\\_destructor](#) `)(void *)`

### 4.8.1 Detailed Description

Vector data structure

### 4.8.2 Field Documentation

#### 4.8.2.1 `size_t ds_vector::current`

Current position

#### 4.8.2.2 `void** ds_vector::data`

Data array

#### 4.8.2.3 `void(* ds_vector::data_destructor)(void *)`

Data destructor function

#### 4.8.2.4 `bool ds_vector::free_on_delete`

'Free on delete' flag

#### 4.8.2.5 `size_t ds_vector::size`

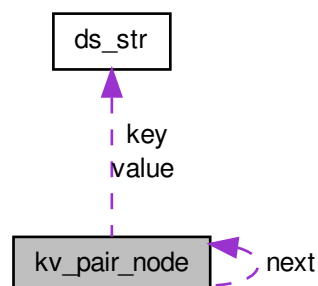
Size of vector

The documentation for this struct was generated from the following file:

- [lib/datastruct/ds\\_vector.c](#)

## 4.9 kv\_pair\_node Struct Reference

Collaboration diagram for kv\_pair\_node:



## Data Fields

- char \* [key](#)
- char \* [value](#)
- struct [kv\\_pair\\_node](#) \* [next](#)
- [ds\\_str](#) [key](#)
- [ds\\_str](#) [value](#)

### 4.9.1 Detailed Description

Structure to hold a key-value pair node

### 4.9.2 Field Documentation

#### 4.9.2.1 `ds_str kv_pair_node::key`

A pointer to the key

#### 4.9.2.2 `char* kv_pair_node::key`

A pointer to the key

#### 4.9.2.3 `struct kv_pair_node * kv_pair_node::next`

A pointer to the next node

#### 4.9.2.4 `ds_str kv_pair_node::value`

A pointer to the value

#### 4.9.2.5 `char* kv_pair_node::value`

A pointer to the value

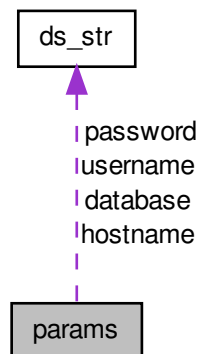
The documentation for this struct was generated from the following files:

- [lib/datastruct/ds\\_map.c](#)
- [lib/datastruct/ds\\_map\\_str.c](#)

## 4.10 params Struct Reference

```
#include <config.h>
```

Collaboration diagram for params:



## Data Fields

- [ds\\_str hostname](#)
- [ds\\_str database](#)
- [ds\\_str username](#)
- [ds\\_str password](#)

### 4.10.1 Detailed Description

Structure to hold database login parameters

### 4.10.2 Field Documentation

#### 4.10.2.1 ds\_str params::database

Database name

#### 4.10.2.2 ds\_str params::hostname

Database hostname

#### 4.10.2.3 ds\_str params::password

Password for database access

#### 4.10.2.4 ds\_str params::username

Username for database access

The documentation for this struct was generated from the following file:

- [config.h](#)

## Chapter 5

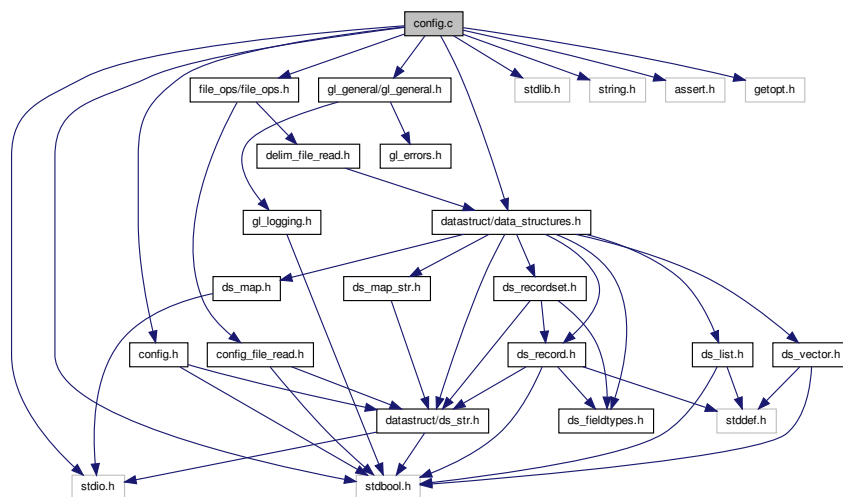
# File Documentation

### 5.1 config.c File Reference

Implementation of program configuration functionality.

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <getopt.h>
#include "config.h"
#include "file_ops/file_ops.h"
#include "datastruct/data_structures.h"
#include "gl_general/gl_general.h"
```

Include dependency graph for config.c:



### Macros

- `#define _XOPEN_SOURCE 500`

## Functions

- struct `params` \* `params_init` (void)  
*Initializes a parameters structure.*
- void `params_free` (struct `params` \*`params`)  
*Frees a parameter structure.*
- bool `get_configuration` (struct `params` \*`params`)  
*Gets parameters from a configuration file.*
- bool `get_cmdline_options` (int `argc`, char \*\*`argv`, struct `params` \*`params`)  
*Gets parameters from the command line.*

### 5.1.1 Detailed Description

Implementation of program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.1.2 Macro Definition Documentation

#### 5.1.2.1 `#define _XOPEN_SOURCE 500`

UNIX feature test macro

### 5.1.3 Function Documentation

#### 5.1.3.1 `bool get_cmdline_options ( int argc, char ** argv, struct params * params )`

Gets parameters from the command line.

#### Parameters

<i>argc</i>	<code>argc</code> as passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

#### Returns

`false` if an unrecognized command line option was specified, `true` otherwise.

#### 5.1.3.2 `bool get_configuration ( struct params * params )`

Gets parameters from a configuration file.



## Parameters

<i>params</i>	A pointer to a parameters structure to populate.
---------------	--

## Returns

`true` on success, `false` otherwise.

## 5.1.3.3 void params\_free ( struct params \* params )

Frees a parameter structure.

## Parameters

<i>params</i>	A pointer to the structure to free.
---------------	-------------------------------------

## 5.1.3.4 struct params\* params\_init ( void ) [read]

Initializes a parameters structure.

## Returns

An initialized parameters structure.

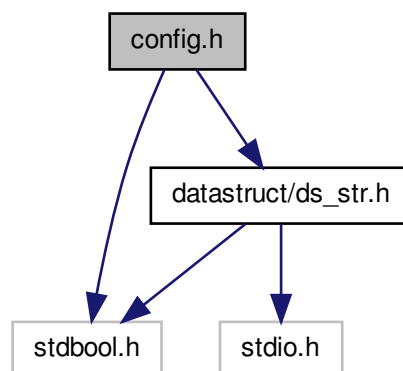
## 5.2 config.h File Reference

Interface to program configuration functionality.

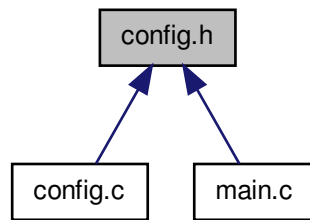
```
#include <stdbool.h>
```

```
#include "datastruct/ds_str.h"
```

Include dependency graph for config.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [params](#)

## Functions

- struct [params](#) \* [params\\_init](#) (void)  
*Initializes a parameters structure.*
- void [params\\_free](#) (struct [params](#) \*[params](#))  
*Frees a parameter structure.*
- bool [get\\_configuration](#) (struct [params](#) \*[params](#))  
*Gets parameters from a configuration file.*
- bool [get\\_cmdline\\_options](#) (int argc, char \*\*argv, struct [params](#) \*[params](#))  
*Gets parameters from the command line.*

### 5.2.1 Detailed Description

Interface to program configuration functionality. Gets program configuration options from the command line and/or a configuration file.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.2.2 Function Documentation

#### 5.2.2.1 bool [get\\_cmdline\\_options](#) ( int *argc*, char \*\* *argv*, struct [params](#) \* *params* )

Gets parameters from the command line.

#### Parameters

<i>argc</i>	<code>argc</code> as passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> as passed to <code>main()</code> .
<i>params</i>	A pointer to a parameters structure to populate.

**Returns**

`false` if an unrecognized command line option was specified, `true` otherwise.

**5.2.2.2 bool get\_configuration ( struct params \* params )**

Gets parameters from a configuration file.

**Parameters**

<i>params</i>	A pointer to a parameters structure to populate.
---------------	--

**Returns**

`true` on success, `false` otherwise.

**5.2.2.3 void params\_free ( struct params \* params )**

Frees a parameter structure.

**Parameters**

<i>params</i>	A pointer to the structure to free.
---------------	-------------------------------------

**5.2.2.4 struct params\* params\_init ( void ) [read]**

Initializes a parameters structure.

**Returns**

An initialized parameters structure.

**5.3 lib/database/database.h File Reference**

User interface to database functionality.

```
#include "datastruct/data_structures.h"
#include "db_connection.h"
#include "db_structure.h"
#include "db_query.h"
#include "db_sampledata.h"
#include "db_reporting.h"
#include "db_users.h"
#include "db_entities.h"
#include "db_nomaccts.h"
#include "db_jes.h"
#include "db_jelines.h"
#include "db_jesrcs.h"
#include "db_standingdata.h"
```

```

graph TD
    DB[Database] --> DBS1[DatabaseSchema]
    DB --> DBS2[DatabaseSchema]
    DB --> DBS3[DatabaseSchema]
    DBS1 --> DBS4[DatabaseSchema]
    DBS2 --> DBS4
    DBS3 --> DBS4
    DBS4 --> DBS5[DatabaseSchema]
    DBS5 --> DBS6[DatabaseSchema]
    DBS6 --> DBS7[DatabaseSchema]
    DBS6 --> DBS8[DatabaseSchema]
    DBS6 --> DBS9[DatabaseSchema]
    DBS6 --> DBS10[DatabaseSchema]
    DBS6 --> DBS11[DatabaseSchema]
    DBS6 --> DBS12[DatabaseSchema]
    DBS6 --> DBS13[DatabaseSchema]
    DBS6 --> DBS14[DatabaseSchema]
    
```

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

```
#include <stdbool.h>
```

```
graph TD; A[lib/database/db_connection.h] --> B[stdbool.h]
```

This graph shows which files directly or indirectly include this file:



## Functions

- bool **db\_connect** (const char \*host, const char \*database, const char \*username, const char \*password)  
*Connects to a database.*
- void **db\_close** (void)  
*Disconnects from a database.*

### 5.4.1 Detailed Description

Interface to database connection functionality. Function implementations are provided by the individual database components.

**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.4.2 Function Documentation

#### 5.4.2.1 bool db\_connect ( const char \* *host*, const char \* *database*, const char \* *username*, const char \* *password* )

Connects to a database.

## Parameters

<i>host</i>	The hostname.
<i>database</i>	The database name.
<i>username</i>	The username with which to connect.
<i>password</i>	The password for the specified user.

## Returns

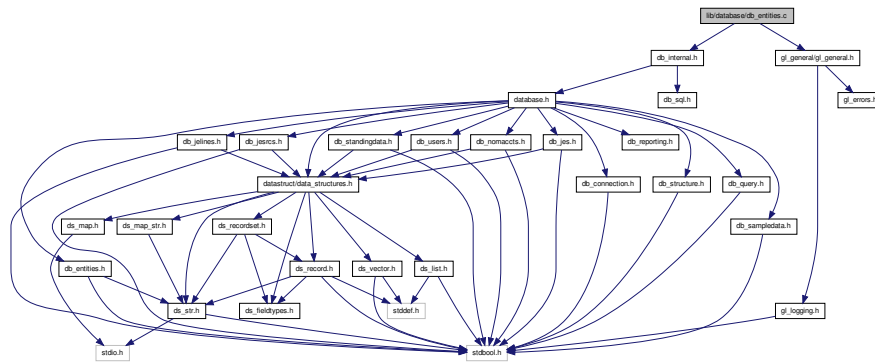
true if the connection was successfully made, false otherwise.

## 5.5 lib/database/db\_entities.c File Reference

### Implementation of entities functionality.

```
#include "db_internal.h"
#include "gl_general/gl_general.h"
```

Include dependency graph for `db_entities.c`:



## Functions

- `bool db_create_entities_table (void)`  
*Creates the entities table in the database.*
- `bool db_drop_entities_table (void)`  
*Drops the entities table in the database.*
- `ds_str db_list_entities_report (void)`  
*Creates a report listing all entities.*

### 5.5.1 Detailed Description

Implementation of entities functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.5.2 Function Documentation

#### 5.5.2.1 `bool db_create_entities_table ( void )`

Creates the entities table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.5.2.2 `bool db_drop_entities_table ( void )`

Drops the entities table in the database.

#### Returns

`true` on success, `false` on failure.

5.5.2.3 `ds_str db_list_entities_report ( void )`

Creates a report listing all entities.

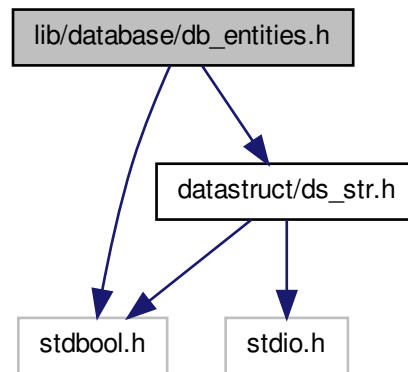
Returns

A `ds_str` containing the report.

## 5.6 lib/database/db\_entities.h File Reference

Interface to entities functionality.

```
#include <stdbool.h>
#include "datastruct/ds_str.h"
Include dependency graph for db_entities.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- bool `db_create_entities_table` (void)  
*Creates the entities table in the database.*
- bool `db_drop_entities_table` (void)  
*Drops the entities table in the database.*
- `ds_str db_list_entities_report` (void)  
*Creates a report listing all entities.*

### 5.6.1 Detailed Description

Interface to entities functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.6.2 Function Documentation

#### 5.6.2.1 `bool db_create_entities_table ( void )`

Creates the entities table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.6.2.2 `bool db_drop_entities_table ( void )`

Drops the entities table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.6.2.3 `ds_str db_list_entities_report ( void )`

Creates a report listing all entities.

#### Returns

A `ds_str` containing the report.

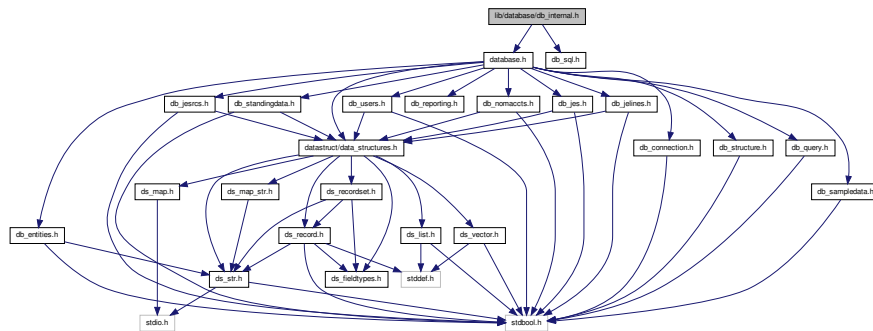
## 5.7 `lib/database/db_internal.h` File Reference

Internal library interface to database functionality.

```
#include "database.h"
#include "db_sql.h"
```



Include dependency graph for db\_internal.h:



This graph shows which files directly or indirectly include this file:



### 5.7.1 Detailed Description

Internal library interface to database functionality. The library interface includes the individual SQL functions which should be encapsulated from the user.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.8 lib/database/db\_jelines.c File Reference

Implementation of journal entries functionality.

```
#include "gl_general/gl_general.h"
#include "db_internal.h"
```



## 5.8.2.3 ds\_str db\_list\_jelines\_report ( void )

Creates a report listing all journal entry lines..

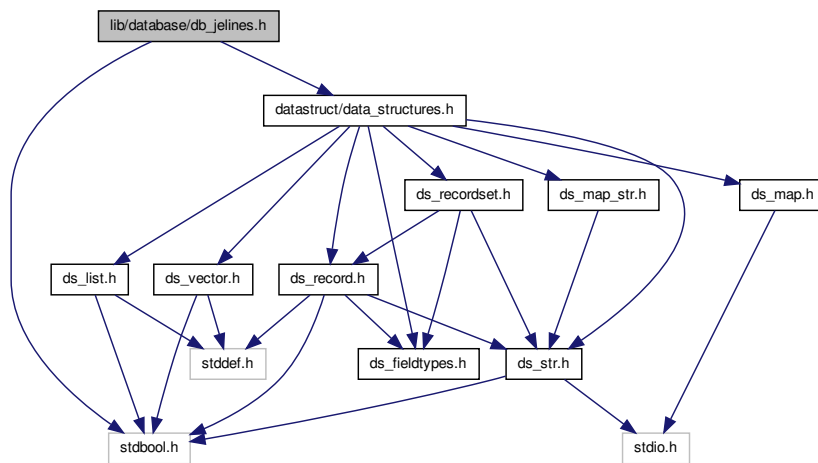
## Returns

A [ds\\_str](#) containing the report.

## 5.9 lib/database/db\_jelines.h File Reference

Interface to journal entry lines functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jelines.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- bool [db\\_create\\_jelines\\_table](#) (void)  
*Creates the journal entry lines table in the database.*
- bool [db\\_drop\\_jelines\\_table](#) (void)  
*Drops the journal entry lines table from the database.*
- [ds\\_str db\\_list\\_jelines\\_report](#) (void)  
*Creates a report listing all journal entry lines..*

### 5.9.1 Detailed Description

Interface to journal entry lines functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.9.2 Function Documentation

#### 5.9.2.1 `bool db_create_jelines_table ( void )`

Creates the journal entry lines table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.9.2.2 `bool db_drop_jelines_table ( void )`

Drops the journal entry lines table from the database.

#### Returns

`true` on success, `false` on failure.

#### 5.9.2.3 `ds_str db_list_jelines_report ( void )`

Creates a report listing all journal entry lines..

#### Returns

A `ds_str` containing the report.

## 5.10 `lib/database/db_jes.c` File Reference

Implementation of journal entries functionality.

```
#include "gl_general/gl_general.h"
#include "db_internal.h"
```

[illegible]

- bool `db_create_jes_table` (void)  
*Creates the journal entries table in the database.*
- bool `db_drop_jes_table` (void)  
*Drops the jes table from the database.*
- `ds_str db_list_jes_report` (void)  
*Creates a report listing all journal entries.*

### Implementation of journal entries functionality.

Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

#### 5.10.2.1 bool db\_create\_jes\_table ( void )

## Returns

true on success, false on failure.

Drops the jes table from the database.

true on success, false on failure.

### 5.10.2.3 `ds_str db_list_jes_report ( void )`

Creates a report listing all journal entries.

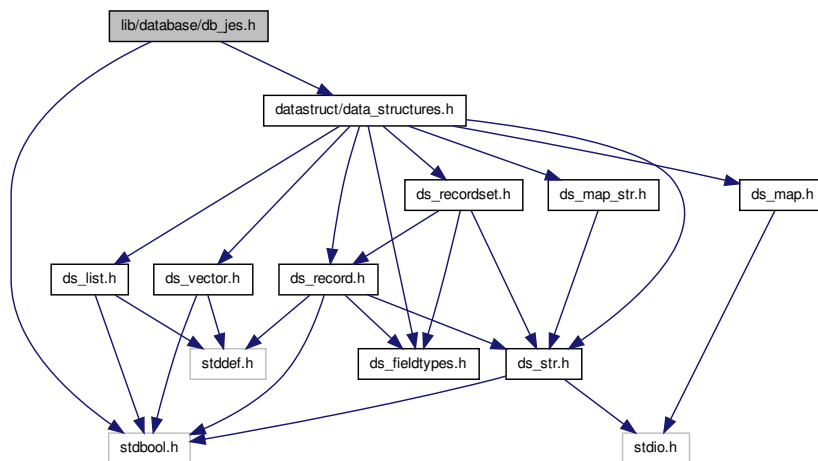
Returns

A `ds_str` containing the report.

## 5.11 `lib/database/db_jes.h` File Reference

Interface to journal entries functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jes.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- `bool db_create_jes_table (void)`  
*Creates the journal entries table in the database.*
- `bool db_drop_jes_table (void)`  
*Drops the jes table from the database.*
- `ds_str db_list_jes_report (void)`  
*Creates a report listing all journal entries.*

### 5.11.1 Detailed Description

Interface to journal entries functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.11.2 Function Documentation

#### 5.11.2.1 `bool db_create_jes_table ( void )`

Creates the journal entries table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.11.2.2 `bool db_drop_jes_table ( void )`

Drops the jes table from the database.

#### Returns

`true` on success, `false` on failure.

#### 5.11.2.3 `ds_str db_list_jes_report ( void )`

Creates a report listing all journal entries.

#### Returns

A `ds_str` containing the report.

## 5.12 lib/database/db\_jesrcs.c File Reference

Implementation of journal entry sources functionality.

```
#include "gl_general/gl_general.h"
#include "db_internal.h"
```



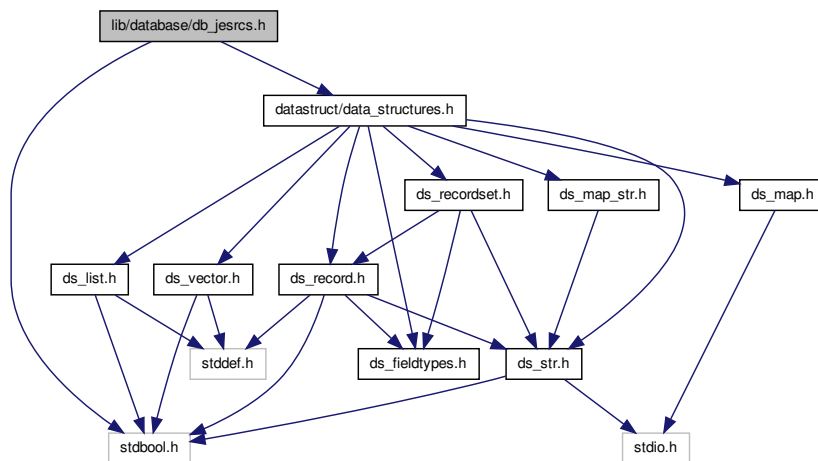


Creates a report listing all journal entry sources.

A `ds_str` containing the report.

Interface to journal entry sources functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_jesrcs.h:
```



This graph shows which files directly or indirectly include this file:



- bool `db_create_jesrcs_table` (void)  
*Creates the JE sources table in the database.*
- bool `db_drop_jesrcs_table` (void)  
*Drops the jesrcs table from the database.*
- ds\_str `db_list_jesrcs_report` (void)  
*Creates a report listing all journal entry sources.*

### 5.13.1 Detailed Description

Interface to journal entry sources functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.13.2 Function Documentation

#### 5.13.2.1 `bool db_create_jesrcs_table ( void )`

Creates the JE sources table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.13.2.2 `bool db_drop_jesrcs_table ( void )`

Drops the jesrcs table from the database.

#### Returns

`true` on success, `false` on failure.

#### 5.13.2.3 `ds_str db_list_jesrcs_report ( void )`

Creates a report listing all journal entry sources.

#### Returns

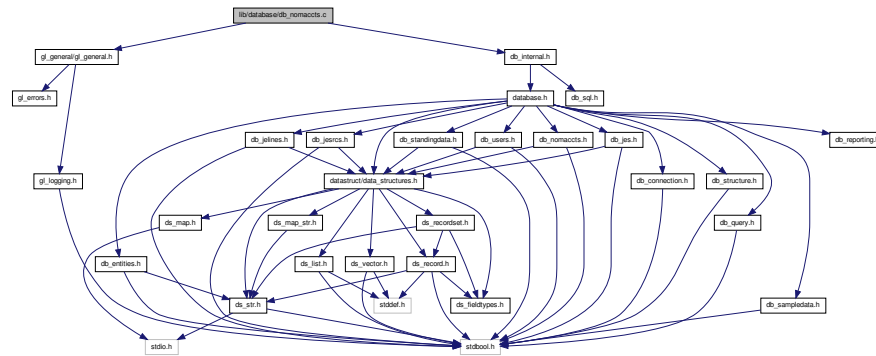
A `ds_str` containing the report.

## 5.14 `lib/database/db_nomaccts.c` File Reference

Implementation of nominal accounts functionality.

```
#include "gl_general/gl_general.h"
#include "db_internal.h"
```

Include dependency graph for db\_nomaccts.c:



## Functions

- bool [db\\_create\\_nomaccts\\_table](#) (void)  
*Creates the nominal accounts table in the database.*
- bool [db\\_drop\\_nomaccts\\_table](#) (void)  
*Drops the nomaccts table from the database.*
- [ds\\_str db\\_list\\_nomaccts\\_report](#) (void)  
*Creates a report listing all nominal accounts.*

### 5.14.1 Detailed Description

Implementation of nominal accounts functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.14.2 Function Documentation

#### 5.14.2.1 bool db\_create\_nomaccts\_table ( void )

Creates the nominal accounts table in the database.

#### Returns

true on success, false on failure.

#### 5.14.2.2 bool db\_drop\_nomaccts\_table ( void )

Drops the nomaccts table from the database.

#### Returns

true on success, false on failure.

### 5.14.2.3 `ds_str db_list_nomaccts_report ( void )`

Creates a report listing all nominal accounts.

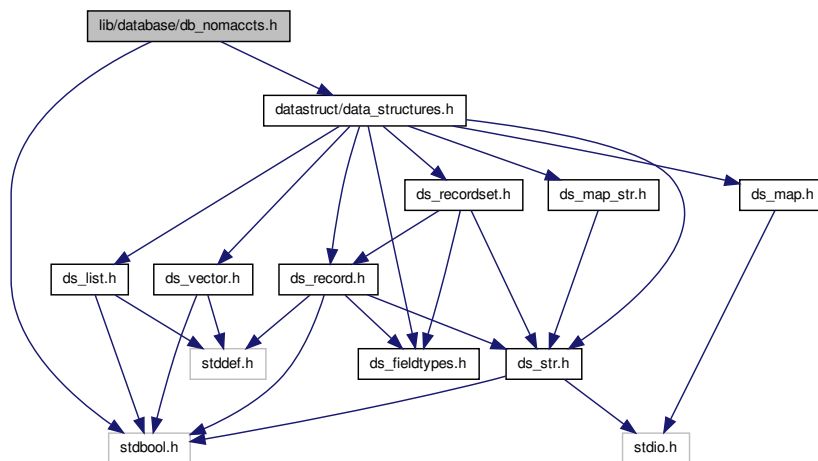
Returns

A `ds_str` containing the report.

## 5.15 `lib/database/db_nomaccts.h` File Reference

Interface to nominal accounts functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_nomaccts.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- bool `db_create_nomaccts_table` (void)  
*Creates the nominal accounts table in the database.*
- bool `db_drop_nomaccts_table` (void)  
*Drops the nominal accounts table from the database.*
- `ds_str db_list_nomaccts_report` (void)  
*Creates a report listing all nominal accounts.*

### 5.15.1 Detailed Description

Interface to nominal accounts functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.15.2 Function Documentation

#### 5.15.2.1 `bool db_create_nomaccts_table ( void )`

Creates the nominal accounts table in the database.

#### Returns

`true` on success, `false` on failure.

#### 5.15.2.2 `bool db_drop_nomaccts_table ( void )`

Drops the nomaccts table from the database.

#### Returns

`true` on success, `false` on failure.

#### 5.15.2.3 `ds_str db_list_nomaccts_report ( void )`

Creates a report listing all nominal accounts.

## Returns

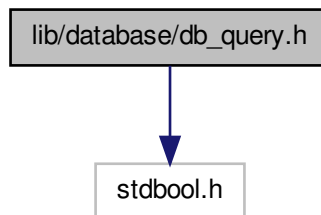
A `ds_str` containing the report.

## 5.16 lib/database/db\_query.h File Reference

Interface to database query functionality.

```
#include <stdbool.h>
```

Include dependency graph for `db_query.h`:



This graph shows which files directly or indirectly include this file:



## Functions

- bool `db_execute_query` (`ds_str` query)  
*Executes an SQL query on the database.*

### 5.16.1 Detailed Description

Interface to database query functionality. Function implementations are provided by the individual database components.

## Author

Paul Griffiths

## Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>







## 5.18.2 Function Documentation

### 5.18.2.1 `ds_recordset db_create_recordset_from_query ( ds_str query )`

Creates a `ds_recordset` from a query.

#### Parameters

<code>query</code>	The SELECT query to run.
--------------------	--------------------------

#### Returns

A `ds_recordset` containing the query result, or `NULL` on failure.

### 5.18.2.2 `ds_str db_create_report_from_query ( ds_str query )`

Creates a text report from a query.

#### Parameters

<code>query</code>	The SELECT query to run.
--------------------	--------------------------

#### Returns

A `ds_str` containing the report, or `NULL` on failure.

### 5.18.2.3 `ds_str db_current_trial_balance_report ( ds_str entity )`

Runs the current trial balance report.

#### Returns

The report.

## 5.19 lib/database/db\_sampledata.c File Reference

Implementation of database sample data functionality.

```
#include <assert.h>
#include "db_internal.h"
#include "file_ops/file_ops.h"
#include "gl_general/gl_general.h"
```

- bool `db_load_sample_data` (void)  
*Loads sample data into the database.*

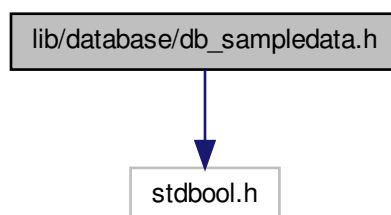
Implementation of database sample data functionality.

Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Interface to database sample data functionality.

Include dependency graph for db\_sampledata.h:



This graph shows which files directly or indirectly include this file:



## Functions

- bool `db_load_sample_data` (void)  
*Loads sample data into the database.*

### 5.20.1 Detailed Description

Interface to database sample data functionality.

**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.21 lib/database/db\_sql.h File Reference

Interface to database specific SQL strings.

This graph shows which files directly or indirectly include this file:



## Functions

- const char \* [db\\_create\\_users\\_table\\_sql](#) (void)  
*Returns the SQL query to create the users table.*
- const char \* [db\\_drop\\_users\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the users table.*
- const char \* [db\\_list\\_users\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list users" report.*
- const char \* [db\\_create\\_entities\\_table\\_sql](#) (void)  
*Returns the SQL query to create the entities table.*
- const char \* [db\\_drop\\_entities\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the entities table.*
- const char \* [db\\_list\\_entities\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list entities" report.*

- `const char * db_create_jes_table_sql (void)`  
*Returns the SQL query to create the journal entries table.*
- `const char * db_drop_jes_table_sql (void)`  
*Returns the SQL query to drop the journal entries table.*
- `const char * db_list_jes_report_sql (void)`  
*Returns the SQL query to run the "list journal entries" report.*
- `const char * db_create_nomaccts_table_sql (void)`  
*Returns the SQL query to create the nominal accounts table.*
- `const char * db_drop_nomaccts_table_sql (void)`  
*Returns the SQL query to drop the nominal accounts table.*
- `const char * db_list_nomaccts_report_sql (void)`  
*Returns the SQL query to run the "list nominal accounts" report.*
- `const char * db_create_jelines_table_sql (void)`  
*Returns the SQL query to create the JE lines table.*
- `const char * db_drop_jelines_table_sql (void)`  
*Returns the SQL query to drop the JE lines table.*
- `const char * db_list_jelines_report_sql (void)`  
*Returns the SQL query to run the "list JE lines" report.*
- `const char * db_current_trial_balance_report_sql (void)`  
*Returns the SQL query to run the "current TB" report.*
- `const char * db_create_jesrcs_table_sql (void)`  
*Returns the SQL query to create the JE sources table.*
- `const char * db_drop_jesrcs_table_sql (void)`  
*Returns the SQL query to drop the JE sources table.*
- `const char * db_list_jesrcs_report_sql (void)`  
*Returns the SQL query to run the "list JE sources" report.*
- `const char * db_create_standingdata_table_sql (void)`  
*Returns the SQL query to create the standing data table.*
- `const char * db_drop_standingdata_table_sql (void)`  
*Returns the SQL query to drop the standing data table.*
- `const char * db_show_standingdata_report_sql (void)`  
*Returns the SQL query to run the "show standing data" report.*

### 5.21.1 Detailed Description

Interface to database specific SQL strings. Function implementations are provided by the individual database components.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.21.2 Function Documentation

### 5.21.2.1 `const char* db.create_entities_table_sql ( void )`

Returns the SQL query to create the entities table.

Returns

The SQL query.

### 5.21.2.2 `const char* db.create_jelines_table_sql ( void )`

Returns the SQL query to create the JE lines table.

Returns

The SQL query.

### 5.21.2.3 `const char* db.create_jes_table_sql ( void )`

Returns the SQL query to create the journal entries table.

Returns

The SQL query.

### 5.21.2.4 `const char* db.create_jesrcs_table_sql ( void )`

Returns the SQL query to create the JE sources table.

Returns

The SQL query.

### 5.21.2.5 `const char* db.create_nomaccts_table_sql ( void )`

Returns the SQL query to create the nominal accounts table.

Returns

The SQL query.

### 5.21.2.6 `const char* db.create_standingdata_table_sql ( void )`

Returns the SQL query to create the standing data table.

Returns

The SQL query.

**5.21.2.7    const char\* db\_create\_users\_table\_sql ( void )**

Returns the SQL query to create the users table.

**Returns**

The SQL query.

**5.21.2.8    const char\* db\_current\_trial\_balance\_report\_sql ( void )**

Returns the SQL query to run the "current TB" report.

**Returns**

The SQL query.

**5.21.2.9    const char\* db\_drop\_entities\_table\_sql ( void )**

Returns the SQL query to drop the entities table.

**Returns**

The SQL query.

**5.21.2.10    const char\* db\_drop\_jelines\_table\_sql ( void )**

Returns the SQL query to drop the JE lines table.

**Returns**

The SQL query.

**5.21.2.11    const char\* db\_drop\_jes\_table\_sql ( void )**

Returns the SQL query to drop the journal entries table.

**Returns**

The SQL query.

**5.21.2.12    const char\* db\_drop\_jesrcs\_table\_sql ( void )**

Returns the SQL query to drop the JE sources table.

**Returns**

The SQL query.

**5.21.2.13    const char\* db\_drop\_nomaccts\_table\_sql ( void )**

Returns the SQL query to drop the nominal accounts table.

**Returns**

The SQL query.

**5.21.2.14    const char\* db\_drop\_standingdata\_table\_sql ( void )**

Returns the SQL query to drop the standing data table.

**Returns**

The SQL query.

**5.21.2.15    const char\* db\_drop\_users\_table\_sql ( void )**

Returns the SQL query to drop the users table.

**Returns**

The SQL query.

**5.21.2.16    const char\* db\_list\_entities\_report\_sql ( void )**

Returns the SQL query to run the "list entities" report.

**Returns**

The SQL query.

**5.21.2.17    const char\* db\_list\_jelines\_report\_sql ( void )**

Returns the SQL query to run the "list JE lines" report.

**Returns**

The SQL query.

**5.21.2.18    const char\* db\_list\_jes\_report\_sql ( void )**

Returns the SQL query to run the "list journal entries" report.

**Returns**

The SQL query.

**5.21.2.19    const char\* db\_list\_jesrcs\_report\_sql ( void )**

Returns the SQL query to run the "list JE sources" report.

**Returns**

The SQL query.

**5.21.2.20    const char\* db\_list\_nomaccts\_report\_sql ( void )**

Returns the SQL query to run the "list nominal accounts" report.

**Returns**

The SQL query.





**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

**5.22.2 Function Documentation****5.22.2.1 `bool db_create_standingdata_table ( void )`**

Creates the standing data table in the database.

**Returns**

`true` on success, `false` on failure.

**5.22.2.2 `bool db_drop_standingdata_table ( void )`**

Drops the standingdata table from the database.

**Returns**

`true` on success, `false` on failure.

**5.22.2.3 `ds_str db_show_standingdata_report ( void )`**

Creates a report showing standing data.



## Author

Paul Griffiths

## Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.23.2 Function Documentation

5.23.2.1 `bool db_create_standingdata_table ( void )`

Creates the standing data table in the database.

## Returns

`true` on success, `false` on failure.

5.23.2.2 `bool db_drop_standingdata_table ( void )`

Drops the standingdata table from the database.

## Returns

`true` on success, `false` on failure.

5.23.2.3 `ds_str db_show_standingdata_report ( void )`

Creates a report showing standing data.

## Returns

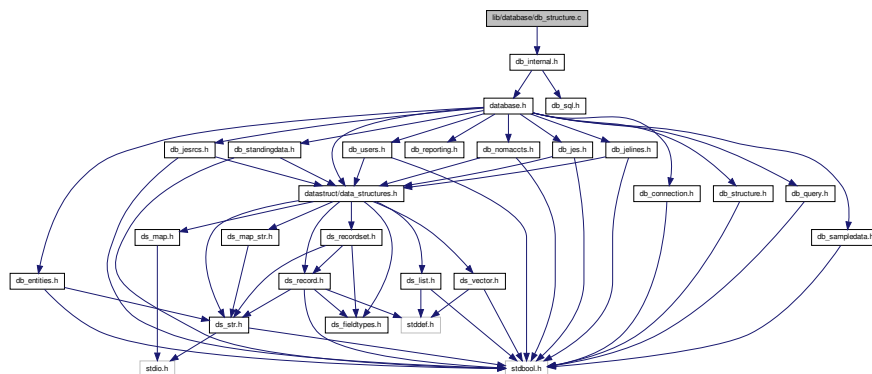
A `ds_str` containing the report.

## 5.24 lib/database/db\_structure.c File Reference

Implementation of database structure functionality.

```
#include "db_internal.h"
```

Include dependency graph for db\_structure.c:



## Functions

- bool [db\\_create\\_database\\_structure](#) (void)

*Creates an empty database structure.*

- bool [db\\_delete\\_database\\_structure](#) (void)

*Deletes the database structure.*

### 5.24.1 Detailed Description

Implementation of database structure functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.24.2 Function Documentation

#### 5.24.2.1 bool [db\\_create\\_database\\_structure](#) ( void )

Creates an empty database structure.

#### Returns

`true` on success, `false` on failure.

#### 5.24.2.2 bool [db\\_delete\\_database\\_structure](#) ( void )

Deletes the database structure.

## Returns

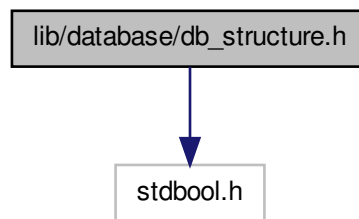
`true` on success, `false` on failure.

## 5.25 lib/database/db\_structure.h File Reference

Interface to database structure functionality.

```
#include <stdbool.h>
```

Include dependency graph for `db_structure.h`:



This graph shows which files directly or indirectly include this file:



## Functions

- bool [db\\_create\\_database\\_structure](#) (void)  
*Creates an empty database structure.*
- bool [db\\_delete\\_database\\_structure](#) (void)  
*Deletes the database structure.*

### 5.25.1 Detailed Description

Interface to database structure functionality.

## Author

Paul Griffiths

## Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>



**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

**5.26.2 Function Documentation****5.26.2.1 bool db\_create\_users\_table ( void )**

Creates the users table in the database.

**Returns**

`true` on success, `false` on failure.

**5.26.2.2 bool db\_drop\_users\_table ( void )**

Drops the users table from the database.

**Returns**

`true` on success, `false` on failure.

**5.26.2.3 ds\_str db\_list\_users\_report ( void )**

Creates a report listing all users.

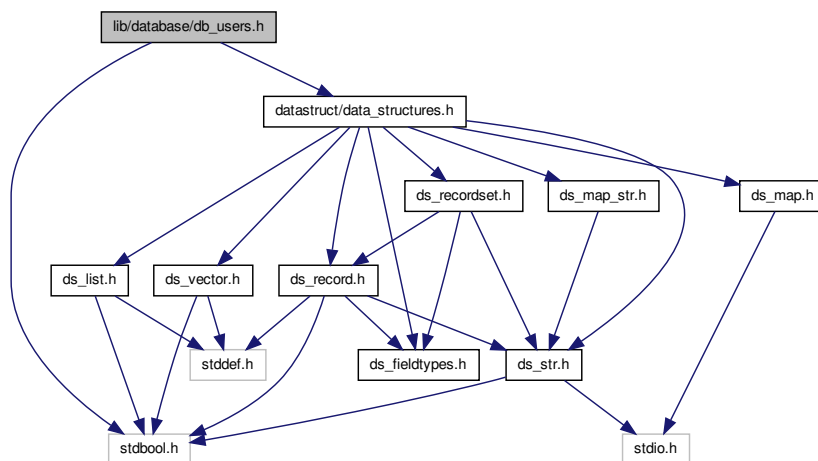
## Returns

A [ds\\_str](#) containing the report.

## 5.27 lib/database/db\_users.h File Reference

Interface to users functionality.

```
#include <stdbool.h>
#include "datastruct/data_structures.h"
Include dependency graph for db_users.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- bool [db\\_create\\_users\\_table](#) (void)  
*Creates the users table in the database.*
- bool [db\\_drop\\_users\\_table](#) (void)  
*Drops the users table from the database.*
- [ds\\_str db\\_list\\_users\\_report](#) (void)  
*Creates a report listing all users.*

### 5.27.1 Detailed Description

Interface to users functionality.



**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

**5.27.2 Function Documentation****5.27.2.1 bool db\_create\_users\_table ( void )**

Creates the users table in the database.

**Returns**

`true` on success, `false` on failure.

**5.27.2.2 bool db\_drop\_users\_table ( void )**

Drops the users table from the database.

**Returns**

`true` on success, `false` on failure.

**5.27.2.3 ds\_str db\_list\_users\_report ( void )**

Creates a report listing all users.

**Returns**

A `ds_str` containing the report.

**5.28 lib/database/dummy/db\_dummy\_create\_entities\_table\_sql.c File Reference**

Returns dummy SQL query to create entities table.

**Functions**

- `const char * db_create_entities_table_sql (void)`  
*Returns the SQL query to create the entities table.*

**5.28.1 Detailed Description**

Returns dummy SQL query to create entities table.

**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.28.2 Function Documentation

### 5.28.2.1 `const char* db_create_entities_table_sql ( void )`

Returns the SQL query to create the entities table.

Returns

The SQL query.

## 5.29 `lib/database/dummy/db_dummy_create_users_table_sql.c` File Reference

Returns dummy SQL query to create users table.

### Functions

- `const char * db_create_users_table_sql (void)`  
*Returns the SQL query to create the users table.*

### 5.29.1 Detailed Description

Returns dummy SQL query to create users table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.29.2 Function Documentation

### 5.29.2.1 `const char* db_create_users_table_sql ( void )`

Returns the SQL query to create the users table.

Returns

The SQL query.

## 5.30 `lib/database/dummy/db_dummy_drop_entities_table_sql.c` File Reference

Returns dummy SQL query to drop entities table.

### Functions

- `const char * db_drop_entities_table_sql (void)`  
*Returns the SQL query to drop the entities table.*

### 5.30.1 Detailed Description

Returns dummy SQL query to drop entities table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.30.2 Function Documentation

#### 5.30.2.1 `const char* db_drop_entities_table_sql ( void )`

Returns the SQL query to drop the entities table.

#### Returns

The SQL query.

## 5.31 lib/database/dummy/db\_dummy\_drop\_users\_table\_sql.c File Reference

Returns dummy SQL query to drop users table.

### Functions

- `const char * db_drop_users_table_sql (void)`  
*Returns the SQL query to drop the users table.*

### 5.31.1 Detailed Description

Returns dummy SQL query to drop users table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.31.2 Function Documentation

#### 5.31.2.1 `const char* db_drop_users_table_sql ( void )`

Returns the SQL query to drop the users table.

#### Returns

The SQL query.



## 5.32.2 Macro Definition Documentation

### 5.32.2.1 `#define _XOPEN_SOURCE 600`

UNIX feature test macro

## 5.32.3 Function Documentation

### 5.32.3.1 `bool db_connect ( const char * host, const char * database, const char * username, const char * password )`

Connects to a database.

#### Parameters

<i>host</i>	The hostname.
<i>database</i>	The database name.
<i>username</i>	The username with which to connect.
<i>password</i>	The password for the specified user.

#### Returns

`true` if the connection was successfully made, `false` otherwise.

### 5.32.3.2 `ds_recordset db_create_recordset_from_query ( ds_str query )`

Creates a [ds\\_recordset](#) from a query.

#### Parameters

<i>query</i>	The SELECT query to run.
--------------	--------------------------

#### Returns

A [ds\\_recordset](#) containing the query result, or `NULL` on failure.

### 5.32.3.3 `bool db_execute_query ( ds_str query )`

Executes an SQL query on the database.

#### Parameters

<i>query</i>	The query to execute.
--------------	-----------------------

#### Returns

`true` if the query was successfully executed, `false` otherwise.

## 5.33 lib/database/dummy/db\_dummy\_list\_entities\_report\_sql.c File Reference

Returns dummy SQL query to create list entities report.

## Functions

- `const char * db_list_entities_report_sql` (void)  
*Returns the SQL query to run the "list entities" report.*

### 5.33.1 Detailed Description

Returns dummy SQL query to create list entities report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.33.2 Function Documentation

#### 5.33.2.1 `const char* db_list_entities_report_sql ( void )`

Returns the SQL query to run the "list entities" report.

#### Returns

The SQL query.

## 5.34 lib/database/dummy/db\_dummy\_list\_users\_report\_sql.c File Reference

Returns dummy SQL query to create list users report.

## Functions

- `const char * db_list_users_report_sql` (void)  
*Returns the SQL query to run the "list users" report.*

### 5.34.1 Detailed Description

Returns dummy SQL query to create list users report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.34.2 Function Documentation

#### 5.34.2.1 `const char* db_list_users_report_sql ( void )`

Returns the SQL query to run the "list users" report.

Returns

The SQL query.

## 5.35 lib/database/mysql/db\_mysql\_create\_entities\_table\_sql.c File Reference

Returns MYSQL SQL query to create entities table.

### Functions

- `const char * db_create_entities_table_sql (void)`  
*Returns the SQL query to create the entities table.*

#### 5.35.1 Detailed Description

Returns MYSQL SQL query to create entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.35.2 Function Documentation

#### 5.35.2.1 `const char* db_create_entities_table_sql ( void )`

Returns the SQL query to create the entities table.

Returns

The SQL query.

## 5.36 lib/database/mysql/db\_mysql\_create\_jelines\_table\_sql.c File Reference

Returns MYSQL SQL query to create journal entry lines table.

### Functions

- `const char * db_create_jelines_table_sql (void)`  
*Returns the SQL query to create the JE lines table.*

### 5.36.1 Detailed Description

Returns MYSQL SQL query to create journal entry lines table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.36.2 Function Documentation

#### 5.36.2.1 `const char* db.create_jelines_table_sql ( void )`

Returns the SQL query to create the JE lines table.

#### Returns

The SQL query.

## 5.37 lib/database/mysql/db\_mysql\_create\_jes\_table\_sql.c File Reference

Returns MYSQL SQL query to create journal entries table.

### Functions

- `const char * db_create_jes_table_sql (void)`  
*Returns the SQL query to create the journal entries table.*

### 5.37.1 Detailed Description

Returns MYSQL SQL query to create journal entries table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.37.2 Function Documentation

#### 5.37.2.1 `const char* db.create_jes_table_sql ( void )`

Returns the SQL query to create the journal entries table.

#### Returns

The SQL query.



## 5.38 lib/database/mysql/db\_mysql\_create\_jesrcs\_table\_sql.c File Reference

Returns MYSQL SQL query to create JE sources table.

### Functions

- const char \* [db\\_create\\_jesrcs\\_table\\_sql](#) (void)  
*Returns the SQL query to create the JE sources table.*

#### 5.38.1 Detailed Description

Returns MYSQL SQL query to create JE sources table.

##### Author

Paul Griffiths

##### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

#### 5.38.2 Function Documentation

##### 5.38.2.1 const char\* db\_create\_jesrcs\_table\_sql ( void )

Returns the SQL query to create the JE sources table.

##### Returns

The SQL query.

## 5.39 lib/database/mysql/db\_mysql\_create\_nomaccts\_table\_sql.c File Reference

Returns MYSQL SQL query to create nominal accounts table.

### Functions

- const char \* [db\\_create\\_nomaccts\\_table\\_sql](#) (void)  
*Returns the SQL query to create the nominal accounts table.*

#### 5.39.1 Detailed Description

Returns MYSQL SQL query to create nominal accounts table.

##### Author

Paul Griffiths

##### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.39.2 Function Documentation

### 5.39.2.1 `const char* db_create_nomaccts_table_sql ( void )`

Returns the SQL query to create the nominal accounts table.

Returns

The SQL query.

## 5.40 `lib/database/mysql/db_mysql_create_standingdata_table_sql.c` File Reference

Returns MYSQL SQL query to create standing data table.

### Functions

- `const char * db_create_standingdata_table_sql (void)`  
*Returns the SQL query to create the standing data table.*

### 5.40.1 Detailed Description

Returns MYSQL SQL query to create standing data table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.40.2 Function Documentation

### 5.40.2.1 `const char* db_create_standingdata_table_sql ( void )`

Returns the SQL query to create the standing data table.

Returns

The SQL query.

## 5.41 `lib/database/mysql/db_mysql_create_users_table_sql.c` File Reference

Returns MYSQL SQL query to create users table.

### Functions

- `const char * db_create_users_table_sql (void)`  
*Returns the SQL query to create the users table.*

### 5.41.1 Detailed Description

Returns MYSQL SQL query to create users table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.41.2 Function Documentation

#### 5.41.2.1 `const char* db.create_users_table_sql ( void )`

Returns the SQL query to create the users table.

#### Returns

The SQL query.

## 5.42 lib/database/mysql/db\_mysql\_current\_trial\_balance\_report\_sql.c File Reference

Returns MYSQL SQL query to create current TB report.

### Functions

- `const char * db_current_trial_balance_report_sql (void)`  
*Returns the SQL query to run the "current TB" report.*

### 5.42.1 Detailed Description

Returns MYSQL SQL query to create current TB report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.42.2 Function Documentation

#### 5.42.2.1 `const char* db.current_trial_balance_report_sql ( void )`

Returns the SQL query to run the "current TB" report.

#### Returns

The SQL query.

## 5.43 lib/database/mysql/db\_mysql\_drop\_entities\_table\_sql.c File Reference

Returns MYSQL SQL query to drop entities table.

### Functions

- const char \* [db\\_drop\\_entities\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the entities table.*

#### 5.43.1 Detailed Description

Returns MYSQL SQL query to drop entities table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

#### 5.43.2 Function Documentation

##### 5.43.2.1 const char\* db\_drop\_entities\_table\_sql ( void )

Returns the SQL query to drop the entities table.

#### Returns

The SQL query.

## 5.44 lib/database/mysql/db\_mysql\_drop\_jelines\_table\_sql.c File Reference

Returns MYSQL SQL query to drop journal entry lines table.

### Functions

- const char \* [db\\_drop\\_jelines\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the JE lines table.*

#### 5.44.1 Detailed Description

Returns MYSQL SQL query to drop journal entry lines table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.44.2 Function Documentation

### 5.44.2.1 `const char* db_drop_jelines_table_sql ( void )`

Returns the SQL query to drop the JE lines table.

Returns

The SQL query.

## 5.45 lib/database/mysql/db\_mysql\_drop\_jes\_table\_sql.c File Reference

Returns MYSQL SQL query to drop entities table.

### Functions

- `const char * db_drop_jes_table_sql (void)`  
*Returns the SQL query to drop the journal entries table.*

### 5.45.1 Detailed Description

Returns MYSQL SQL query to drop entities table.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.45.2 Function Documentation

### 5.45.2.1 `const char* db_drop_jes_table_sql ( void )`

Returns the SQL query to drop the journal entries table.

Returns

The SQL query.

## 5.46 lib/database/mysql/db\_mysql\_drop\_jesrcs\_table\_sql.c File Reference

Returns MYSQL SQL query to drop JE sources table.

### Functions

- `const char * db_drop_jesrcs_table_sql (void)`  
*Returns the SQL query to drop the JE sources table.*

### 5.46.1 Detailed Description

Returns MYSQL SQL query to drop JE sources table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.46.2 Function Documentation

#### 5.46.2.1 `const char* db_drop_jesrcs_table_sql ( void )`

Returns the SQL query to drop the JE sources table.

#### Returns

The SQL query.

## 5.47 lib/database/mysql/db\_mysql\_drop\_nomaccts\_table\_sql.c File Reference

Returns MYSQL SQL query to drop nominal accounts table.

### Functions

- `const char * db_drop_nomaccts_table_sql (void)`  
*Returns the SQL query to drop the nominal accounts table.*

### 5.47.1 Detailed Description

Returns MYSQL SQL query to drop nominal accounts table.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.47.2 Function Documentation

#### 5.47.2.1 `const char* db_drop_nomaccts_table_sql ( void )`

Returns the SQL query to drop the nominal accounts table.

#### Returns

The SQL query.

## 5.48 lib/database/mysql/db\_mysql\_drop\_standingdata\_table\_sql.c File Reference

Returns MYSQL SQL query to drop standing data table.

### Functions

- const char \* [db\\_drop\\_standingdata\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the standing data table.*

#### 5.48.1 Detailed Description

Returns MYSQL SQL query to drop standing data table.

##### Author

Paul Griffiths

##### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

#### 5.48.2 Function Documentation

##### 5.48.2.1 const char\* db\_drop\_standingdata\_table\_sql ( void )

Returns the SQL query to drop the standing data table.

##### Returns

The SQL query.

## 5.49 lib/database/mysql/db\_mysql\_drop\_users\_table\_sql.c File Reference

Returns MYSQL SQL query to drop users table.

### Functions

- const char \* [db\\_drop\\_users\\_table\\_sql](#) (void)  
*Returns the SQL query to drop the users table.*

#### 5.49.1 Detailed Description

Returns MYSQL SQL query to drop users table.

##### Author

Paul Griffiths

##### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>





### 5.50.1 Detailed Description

Implementation of MYSQL database functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.50.2 Function Documentation

#### 5.50.2.1 `bool db_connect ( const char * host, const char * database, const char * username, const char * password )`

Connects to a database.

##### Parameters

<i>host</i>	The hostname.
<i>database</i>	The database name.
<i>username</i>	The username with which to connect.
<i>password</i>	The password for the specified user.

##### Returns

`true` if the connection was successfully made, `false` otherwise.

#### 5.50.2.2 `ds_recordset db_create_recordset_from_query ( ds_str query )`

Creates a [ds\\_recordset](#) from a query.

##### Parameters

<i>query</i>	The SELECT query to run.
--------------	--------------------------

##### Returns

A [ds\\_recordset](#) containing the query result, or `NULL` on failure.

#### 5.50.2.3 `bool db_execute_query ( ds_str query )`

Executes an SQL query on the database.

##### Parameters

<i>query</i>	The query to execute.
--------------	-----------------------

##### Returns

`true` if the query was successfully executed, `false` otherwise.

### 5.50.3 Variable Documentation

#### 5.50.3.1 MYSQL\* conn\_mss = NULL

MYSQL connection object.

#### 5.50.3.2 MYSQL\* main\_mss = NULL

MYSQL initialization object.

## 5.51 lib/database/mysql/db\_mysql\_list\_entities\_report\_sql.c File Reference

Returns MYSQL SQL query to create list entities report.

### Functions

- const char \* [db\\_list\\_entities\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list entities" report.*

#### 5.51.1 Detailed Description

Returns MYSQL SQL query to create list entities report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

#### 5.51.2 Function Documentation

##### 5.51.2.1 const char\* db\_list\_entities\_report\_sql ( void )

Returns the SQL query to run the "list entities" report.

#### Returns

The SQL query.

## 5.52 lib/database/mysql/db\_mysql\_list\_jelines\_report\_sql.c File Reference

Returns MYSQL SQL query to create JE lines report.

### Functions

- const char \* [db\\_list\\_jelines\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list JE lines" report.*

### 5.52.1 Detailed Description

Returns MYSQL SQL query to create JE lines report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.52.2 Function Documentation

#### 5.52.2.1 `const char* db_list_jelines_report_sql ( void )`

Returns the SQL query to run the "list JE lines" report.

#### Returns

The SQL query.

## 5.53 lib/database/mysql/db\_mysql\_list\_jes\_report\_sql.c File Reference

Returns MYSQL SQL query to create journal entries report.

### Functions

- `const char * db_list_jes_report_sql (void)`  
*Returns the SQL query to run the "list journal entries" report.*

### 5.53.1 Detailed Description

Returns MYSQL SQL query to create journal entries report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.53.2 Function Documentation

#### 5.53.2.1 `const char* db_list_jes_report_sql ( void )`

Returns the SQL query to run the "list journal entries" report.

#### Returns

The SQL query.

## 5.54 lib/database/mysql/db\_mysql\_list\_jesrcs\_report\_sql.c File Reference

Returns MYSQL SQL query to create JE sources report.

### Functions

- const char \* [db\\_list\\_jesrcs\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list JE sources" report.*

### 5.54.1 Detailed Description

Returns MYSQL SQL query to create JE sources report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.54.2 Function Documentation

#### 5.54.2.1 const char\* db\_list\_jesrcs\_report\_sql ( void )

Returns the SQL query to run the "list JE sources" report.

#### Returns

The SQL query.

## 5.55 lib/database/mysql/db\_mysql\_list\_nomaccts\_report\_sql.c File Reference

Returns MYSQL SQL query to create list nominal accounts report.

### Functions

- const char \* [db\\_list\\_nomaccts\\_report\\_sql](#) (void)  
*Returns the SQL query to run the "list nominal accounts" report.*

### 5.55.1 Detailed Description

Returns MYSQL SQL query to create list nominal accounts report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.55.2 Function Documentation

### 5.55.2.1 `const char* db_list_nomaccts_report_sql ( void )`

Returns the SQL query to run the "list nominal accounts" report.

Returns

The SQL query.

## 5.56 lib/database/mysql/db\_mysql\_list\_users\_report\_sql.c File Reference

Returns MYSQL SQL query to create list users report.

### Functions

- `const char * db_list_users_report_sql (void)`  
*Returns the SQL query to run the "list users" report.*

### 5.56.1 Detailed Description

Returns MYSQL SQL query to create list users report.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.56.2 Function Documentation

### 5.56.2.1 `const char* db_list_users_report_sql ( void )`

Returns the SQL query to run the "list users" report.

Returns

The SQL query.

## 5.57 lib/database/mysql/db\_mysql\_show\_standingdata\_report\_sql.c File Reference

Returns MYSQL SQL query to create show standing data report.

### Functions

- `const char * db_show_standingdata_report_sql (void)`  
*Returns the SQL query to run the "show standing data" report.*

### 5.57.1 Detailed Description

Returns MYSQL SQL query to create show standing data report.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.57.2 Function Documentation

#### 5.57.2.1 `const char* db_show_standingdata_report_sql ( void )`

Returns the SQL query to run the "show standing data" report.

#### Returns

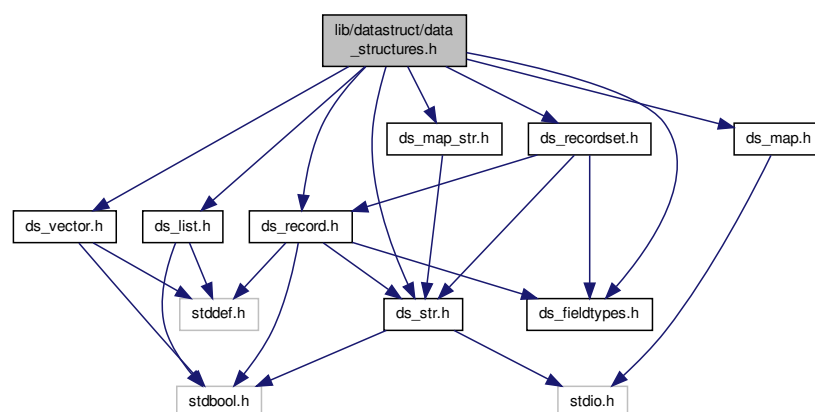
The SQL query.

## 5.58 lib/datastruct/data\_structures.h File Reference

Interface to data structures.

```
#include "ds_list.h"
#include "ds_vector.h"
#include "ds_str.h"
#include "ds_map.h"
#include "ds_map_str.h"
#include "ds_fieldtypes.h"
#include "ds_record.h"
#include "ds_recordset.h"
```

Include dependency graph for data\_structures.h:



This graph shows which files directly or indirectly include this file:



### 5.58.1 Detailed Description

Interface to data structures.

#### Author

Paul Griffiths

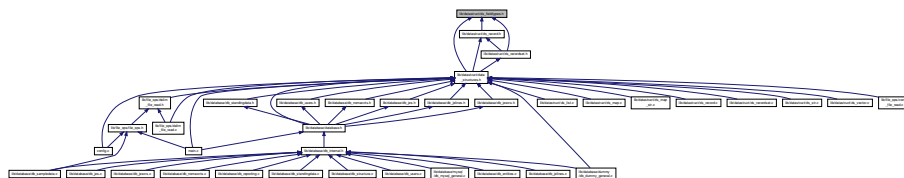
#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.59 lib/datastruct/ds\_fieldtypes.h File Reference

Record field types enumeration.

This graph shows which files directly or indirectly include this file:



### Enumerations

- enum `ds_field_types` { `DS_FIELD_STRING`, `DS_FIELD_INT`, `DS_FIELD_BOOLEAN`, `DS_FIELD_DOUBLE` }

### 5.59.1 Detailed Description

Record field types enumeration.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.59.2 Enumeration Type Documentation

### 5.59.2.1 enum ds\_field\_types

Enumeration for field type

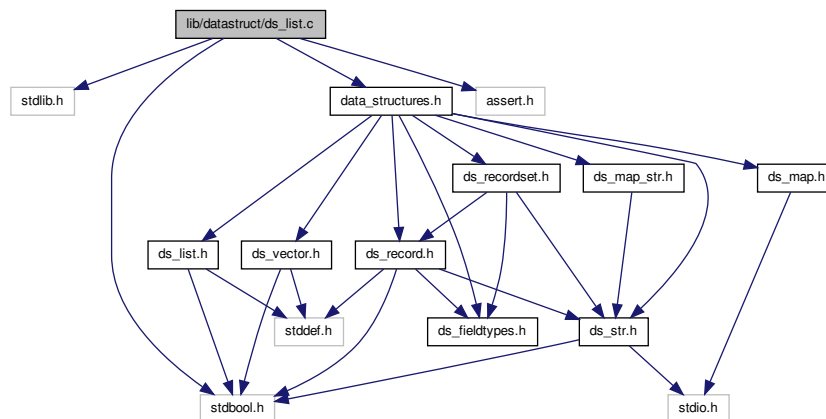
Enumerator:

**DS\_FIELD\_STRING** Field is string type  
**DS\_FIELD\_INT** Field is integer type  
**DS\_FIELD\_BOOLEAN** Field is boolean type  
**DS\_FIELD\_DOUBLE** Field is double type

## 5.60 lib/datastruct/ds\_list.c File Reference

Implementation of generic doubly-linked list data structure.

```
#include <stdlib.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_list.c:
```



## Data Structures

- struct [ds\\_list\\_element](#)
- struct [ds\\_list](#)

## Functions

- [ds\\_list ds\\_list\\_create](#) (const bool free\_on\_delete, void(\*destructor)(void \*))  
*Creates a new list.*
- void [ds\\_list\\_destroy](#) (ds\_list list)  
*Destroys a list and frees any associated resources.*
- void [ds\\_list\\_destructor](#) (void \*list)  
*A list destructor function.*



- `ds_list ds_list_append` (`ds_list` list, void \*data)  
*Appends an element to a list.*
- void `ds_list_remove_tail` (`ds_list` list)  
*Removes the last element of a list.*
- void `ds_list_remove_all` (`ds_list` list)  
*Removes all the elements from a list.*
- void \* `ds_list_element` (`ds_list` list, const size\_t index)  
*Retrieves the data at a specified index.*
- size\_t `ds_list_length` (`ds_list` list)  
*Returns the number of elements in a list.*
- bool `ds_list_is_empty` (`ds_list` list)  
*Checks if a list is empty.*
- void `ds_list_seek_start` (`ds_list` list)  
*Sets the current element to the first element of a list.*
- void `ds_list_seek_end` (`ds_list` list)  
*Sets the current element to the last element of a list.*
- void \* `ds_list_get_next_data` (`ds_list` list)  
*Returns the next element of the list.*
- void \* `ds_list_get_prev_data` (`ds_list` list)  
*Returns the previous element of the list.*

### 5.60.1 Detailed Description

Implementation of generic doubly-linked list data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.60.2 Function Documentation

#### 5.60.2.1 `ds_list ds_list_append ( ds_list list, void * element )`

Appends an element to a list.

#### Parameters

<i>list</i>	The list to which to append.
<i>element</i>	The element to append.

#### Returns

The same list, or `NULL` on failure.

#### 5.60.2.2 `ds_list ds_list_create ( const bool free_on_delete, void(*) (void *) destructor )`

Creates a new list.

## Parameters

<i>free_on_delete</i>	Set to <code>true</code> if the list elements should be destroyed when removed from the list, and when the list itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the list.
<i>destructor</i>	Pointer to a destructor function to use for destroying the list elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

## Returns

A newly created list, or `NULL` on failure.

5.60.2.3 `void ds_list_destroy ( ds_list list )`

Destroys a list and frees any associated resources.

## Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.60.2.4 `void ds_list_destructor ( void * list )`

A list destructor function.

This function may be passed to `ds_list_create()` when creating a list of lists. It calls `ds_list_destroy()`, but the parameter of `ds_list_destroy()` is not compatible with the function signature expected by `ds_list_create()`, so this function provides an appropriate interface.

## Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

5.60.2.5 `void* ds_list_element ( ds_list list, const size_t index )`

Retrieves the data at a specified index.

## Parameters

<i>list</i>	The list from which to retrieve.
<i>index</i>	The index of the desired element.

## Returns

A pointer to the data, or `NULL` if the index is out of range.

5.60.2.6 `void* ds_list_get_next_data ( ds_list list )`

Returns the next element of the list.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

## Parameters

<i>list</i>	The list.
-------------	-----------

## Returns

A pointer to the next element, or `NULL` if the end of the list has been reached.

**5.60.2.7 void\* ds\_list\_get\_prev\_data ( ds\_list list )**

Returns the previous element of the list.

This function returns the data of the "current element", and decrements the current element pointer. Subsequent calls to this function will return successively earlier elements.

## Parameters

<i>list</i>	The list.
-------------	-----------

## Returns

A pointer to the previous element, or `NULL` if the start of the list has been reached.

**5.60.2.8 bool ds\_list\_is\_empty ( ds\_list list )**

Checks if a list is empty.

## Parameters

<i>list</i>	The list to check.
-------------	--------------------

## Returns

`true` if the list is empty, `false` otherwise.

**5.60.2.9 size\_t ds\_list\_length ( ds\_list list )**

Returns the number of elements in a list.

## Parameters

<i>list</i>	The list.
-------------	-----------

## Returns

The number of elements in the list.

**5.60.2.10 void ds\_list\_remove\_all ( ds\_list list )**

Removes all the elements from a list.

## Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------



## Typedefs

- typedef struct [ds\\_list](#) \* [ds\\_list](#)

## Functions

- [ds\\_list ds\\_list\\_create](#) (const bool free\_on\_delete, void(\*destructor)(void \*))  
*Creates a new list.*
- void [ds\\_list\\_destroy](#) ([ds\\_list](#) list)  
*Destroys a list and frees any associated resources.*
- void [ds\\_list\\_destructor](#) (void \*list)  
*A list destructor function.*
- [ds\\_list ds\\_list\\_append](#) ([ds\\_list](#) list, void \*element)  
*Appends an element to a list.*
- void [ds\\_list\\_remove\\_tail](#) ([ds\\_list](#) list)  
*Removes the last element of a list.*
- void [ds\\_list\\_remove\\_all](#) ([ds\\_list](#) list)  
*Removes all the elements from a list.*
- void \* [ds\\_list\\_element](#) ([ds\\_list](#) list, const size\_t index)  
*Retrieves the data at a specified index.*
- size\_t [ds\\_list\\_length](#) ([ds\\_list](#) list)  
*Returns the number of elements in a list.*
- bool [ds\\_list\\_is\\_empty](#) ([ds\\_list](#) list)  
*Checks if a list is empty.*
- void [ds\\_list\\_seek\\_start](#) ([ds\\_list](#) list)  
*Sets the current element to the first element of a list.*
- void [ds\\_list\\_seek\\_end](#) ([ds\\_list](#) list)  
*Sets the current element to the last element of a list.*
- void \* [ds\\_list\\_get\\_next\\_data](#) ([ds\\_list](#) list)  
*Returns the next element of the list.*
- void \* [ds\\_list\\_get\\_prev\\_data](#) ([ds\\_list](#) list)  
*Returns the previous element of the list.*

### 5.61.1 Detailed Description

Interface to generic doubly-linked list data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.61.2 Typedef Documentation

#### 5.61.2.1 typedef struct [ds\\_list](#)\* [ds\\_list](#)

Typedef for opaque list datatype

### 5.61.3 Function Documentation

#### 5.61.3.1 `ds_list ds_list_append ( ds_list list, void * element )`

Appends an element to a list.

##### Parameters

<i>list</i>	The list to which to append.
<i>element</i>	The element to append.

##### Returns

The same list, or `NULL` on failure.

#### 5.61.3.2 `ds_list ds_list_create ( const bool free_on_delete, void(*)(void *) destructor )`

Creates a new list.

##### Parameters

<i>free_on_delete</i>	Set to <code>true</code> if the list elements should be destroyed when removed from the list, and when the list itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the list.
<i>destructor</i>	Pointer to a destructor function to use for destroying the list elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

##### Returns

A newly created list, or `NULL` on failure.

#### 5.61.3.3 `void ds_list_destroy ( ds_list list )`

Destroys a list and frees any associated resources.

##### Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

#### 5.61.3.4 `void ds_list_destructor ( void * list )`

A list destructor function.

This function may be passed to `ds_list_create()` when creating a list of lists. It calls `ds_list_destroy()`, but the parameter of `ds_list_destroy()` is not compatible with the function signature expected by `ds_list_create()`, so this function provides an appropriate interface.

##### Parameters

<i>list</i>	The list to destroy.
-------------	----------------------

**5.61.3.5 void\* ds\_list\_element ( ds\_list list, const size\_t index )**

Retrieves the data at a specified index.

**Parameters**

<i>list</i>	The list from which to retrieve.
<i>index</i>	The index of the desired element.

**Returns**

A pointer to the data, or `NULL` if the index is out of range.

**5.61.3.6 void\* ds\_list\_get\_next\_data ( ds\_list list )**

Returns the next element of the list.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

**Parameters**

<i>list</i>	The list.
-------------	-----------

**Returns**

A pointer to the next element, or `NULL` if the end of the list has been reached.

**5.61.3.7 void\* ds\_list\_get\_prev\_data ( ds\_list list )**

Returns the previous element of the list.

This function returns the data of the "current element", and decrements the current element pointer. Subsequent calls to this function will return successively earlier elements.

**Parameters**

<i>list</i>	The list.
-------------	-----------

**Returns**

A pointer to the previous element, or `NULL` if the start of the list has been reached.

**5.61.3.8 bool ds\_list\_is\_empty ( ds\_list list )**

Checks if a list is empty.

**Parameters**

<i>list</i>	The list to check.
-------------	--------------------

**Returns**

`true` if the list is empty, `false` otherwise.

#### 5.61.3.9 `size_t ds_list_length ( ds_list list )`

Returns the number of elements in a list.

##### Parameters

<i>list</i>	The list.
-------------	-----------

##### Returns

The number of elements in the list.

#### 5.61.3.10 `void ds_list_remove_all ( ds_list list )`

Removes all the elements from a list.

##### Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

#### 5.61.3.11 `void ds_list_remove_tail ( ds_list list )`

Removes the last element of a list.

##### Parameters

<i>list</i>	The list from which to remove.
-------------	--------------------------------

#### 5.61.3.12 `void ds_list_seek_end ( ds_list list )`

Sets the current element to the last element of a list.

##### Parameters

<i>list</i>	The list.
-------------	-----------

#### 5.61.3.13 `void ds_list_seek_start ( ds_list list )`

Sets the current element to the first element of a list.

##### Parameters

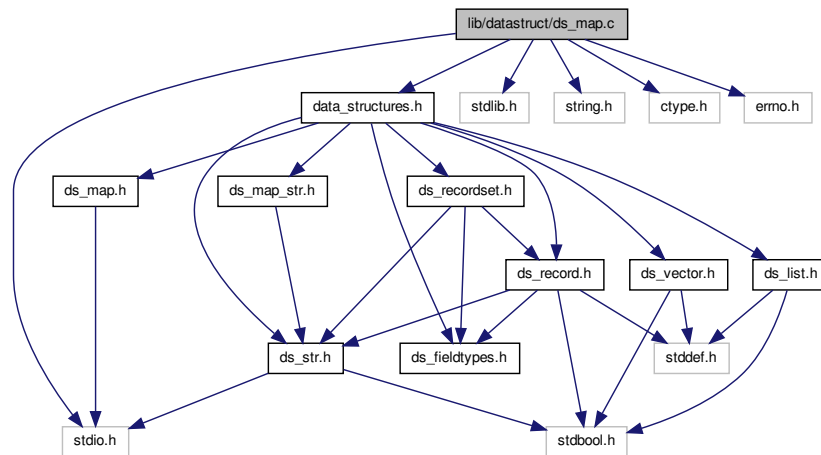
<i>list</i>	The list.
-------------	-----------

## 5.62 `lib/datastruct/ds_map.c` File Reference

Implementation of string-string hash map data structure.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
#include "data_structures.h"
Include dependency graph for ds_map.c:
```



## Data Structures

- struct [kv\\_pair\\_node](#)
- struct [ds\\_map](#)

## Macros

- `#define` [\\_POSIX\\_C\\_SOURCE](#) 200809L  
*Enables POSIX library functions.*

## Functions

- [ds\\_map ds\\_map\\_init](#) (const size\_t hash\_size)  
*Initializes a hash map.*
- void [ds\\_map\\_destroy](#) (ds\_map map)  
*Destroys a hash map.*
- const char \* [ds\\_map\\_get\\_value](#) (ds\_map map, const char \*key)  
*Retrieves a value associated with a key in the map.*
- void [ds\\_map\\_insert](#) (ds\_map map, const char \*key, const char \*value)  
*Inserts a key-value pair into a map.*
- void [ds\\_map\\_print\\_all](#) (ds\_map map, FILE \*outfile)  
*Prints all the key-value pairs in a map to stdout.*

### 5.62.1 Detailed Description

Implementation of string-string hash map data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.62.2 Function Documentation

#### 5.62.2.1 void ds\_map\_destroy ( ds\_map map )

Destroys a hash map.

##### Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

#### 5.62.2.2 const char\* ds\_map\_get\_value ( ds\_map map, const char \* key )

Retrieves a value associated with a key in the map.

##### Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

##### Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

#### 5.62.2.3 ds\_map ds\_map\_init ( const size\_t hash\_size )

Initializes a hash map.

##### Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

##### Returns

A reference to the newly-created hash map.

#### 5.62.2.4 void ds\_map\_insert ( ds\_map map, const char \* key, const char \* value )

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

## Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

## 5.62.2.5 void ds\_map\_print\_all ( ds\_map map, FILE \* outfile )

Prints all the key-value pairs in a map to stdout.

## Parameters

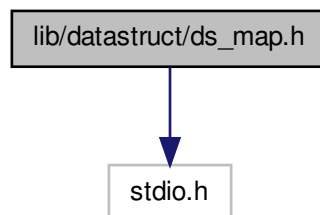
<i>map</i>	A reference to the map.
<i>outfile</i>	A FILE pointer to which to print the output.

## 5.63 lib/datastruct/ds\_map.h File Reference

Interface to string-string hash map data structure.

```
#include <stdio.h>
```

Include dependency graph for ds\_map.h:



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef struct [ds\\_map](#) \* [ds\\_map](#)

## Functions

- [ds\\_map ds\\_map\\_init](#) (const size\_t hash\_size)  
*Initializes a hash map.*

- void `ds_map_destroy` (`ds_map` map)  
*Destroys a hash map.*
- const char \* `ds_map_get_value` (`ds_map` map, const char \*key)  
*Retrieves a value associated with a key in the map.*
- void `ds_map_insert` (`ds_map` map, const char \*key, const char \*value)  
*Inserts a key-value pair into a map.*
- void `ds_map_print_all` (`ds_map` map, FILE \*outfile)  
*Prints all the key-value pairs in a map to stdout.*

### 5.63.1 Detailed Description

Interface to string-string hash map data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.63.2 Typedef Documentation

#### 5.63.2.1 typedef struct ds\_map\* ds\_map

Opaque data type for hash map

### 5.63.3 Function Documentation

#### 5.63.3.1 void ds\_map\_destroy ( ds\_map map )

Destroys a hash map.

#### Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

#### 5.63.3.2 const char\* ds\_map\_get\_value ( ds\_map map, const char \* key )

Retrieves a value associated with a key in the map.

#### Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

#### Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

**5.63.3.3 ds\_map ds\_map\_init ( const size\_t hash\_size )**

Initializes a hash map.

**Parameters**

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

**Returns**

A reference to the newly-created hash map.

**5.63.3.4 void ds\_map\_insert ( ds\_map map, const char \* key, const char \* value )**

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

**Parameters**

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

**5.63.3.5 void ds\_map\_print\_all ( ds\_map map, FILE \* outfile )**

Prints all the key-value pairs in a map to stdout.

**Parameters**

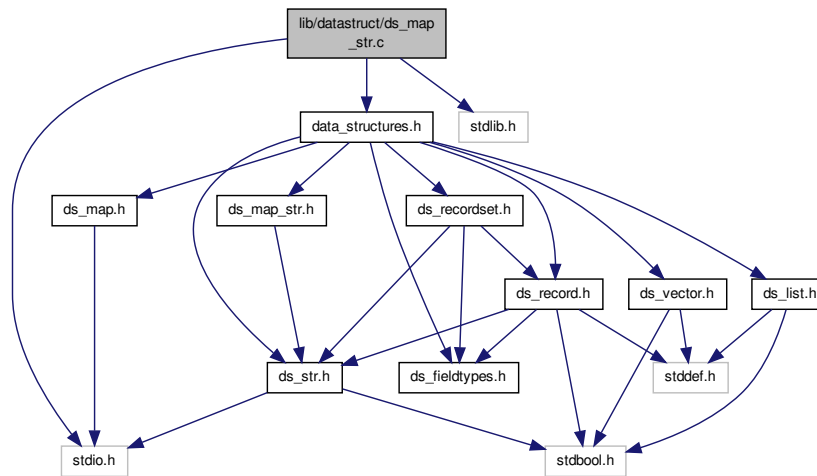
<i>map</i>	A reference to the map.
<i>outfile</i>	A FILE pointer to which to print the output.

**5.64 lib/datastruct/ds\_map\_str.c File Reference**

Implementation of string-string hash map data structure.

```
#include <stdio.h>
#include <stdlib.h>
#include "data_structures.h"
```

Include dependency graph for `ds_map_str.c`:



## Data Structures

- struct [kv\\_pair\\_node](#)
- struct [ds\\_map\\_str](#)

## Functions

- [ds\\_map\\_str ds\\_map\\_str\\_init](#) (const size\_t hash\_size)  
*Initializes a hash map.*
- void [ds\\_map\\_str\\_destroy](#) (ds\_map\_str map)  
*Destroys a hash map.*
- [ds\\_str ds\\_map\\_str\\_get\\_value](#) (ds\_map\_str map, ds\_str key)  
*Retrieves a value associated with a key in the map.*
- void [ds\\_map\\_str\\_insert](#) (ds\_map\_str map, ds\_str key, ds\_str value)  
*Inserts a key-value pair into a map.*

### 5.64.1 Detailed Description

Implementation of string-string hash map data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.64.2 Function Documentation

### 5.64.2.1 void ds\_map\_str\_destroy ( ds\_map\_str map )

Destroys a hash map.

#### Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

### 5.64.2.2 ds\_str ds\_map\_str\_get\_value ( ds\_map\_str map, ds\_str key )

Retrieves a value associated with a key in the map.

#### Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

#### Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

### 5.64.2.3 ds\_map\_str ds\_map\_str\_init ( const size\_t hash\_size )

Initializes a hash map.

#### Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

#### Returns

A reference to the newly-created hash map.

### 5.64.2.4 void ds\_map\_str\_insert ( ds\_map\_str map, ds\_str key, ds\_str value )

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

#### Parameters

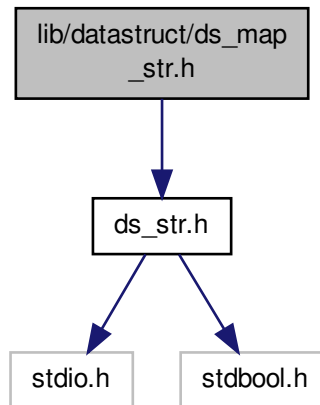
<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

## 5.65 lib/datastruct/ds\_map\_str.h File Reference

Interface to string-string hash map data structure.

```
#include "ds_str.h"
```

Include dependency graph for ds\_map\_str.h:



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef struct `ds_map_str` \* `ds_map_str`

## Functions

- `ds_map_str ds_map_str_init` (const size\_t hash\_size)  
*Initializes a hash map.*
- void `ds_map_str_destroy` (`ds_map_str` map)  
*Destroys a hash map.*
- `ds_str ds_map_str_get_value` (`ds_map_str` map, `ds_str` key)  
*Retrieves a value associated with a key in the map.*
- void `ds_map_str_insert` (`ds_map_str` map, `ds_str` key, `ds_str` value)  
*Inserts a key-value pair into a map.*

### 5.65.1 Detailed Description

Interface to string-string hash map data structure.



## Author

Paul Griffiths

## Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.65.2 Typedef Documentation

## 5.65.2.1 typedef struct ds\_map\_str\* ds\_map\_str

Opaque data type for hash map

## 5.65.3 Function Documentation

## 5.65.3.1 void ds\_map\_str\_destroy ( ds\_map\_str map )

Destroys a hash map.

## Parameters

<i>map</i>	A reference to the map to destroy.
------------	------------------------------------

## 5.65.3.2 ds\_str ds\_map\_str\_get\_value ( ds\_map\_str map, ds\_str key )

Retrieves a value associated with a key in the map.

## Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.

## Returns

A pointer to the value associated with the key, or `NULL` if the key is not in the map. The caller should not modify the string to which this pointer points.

## 5.65.3.3 ds\_map\_str ds\_map\_str\_init ( const size\_t hash\_size )

Initializes a hash map.

## Parameters

<i>hash_size</i>	The number of possible hash values.
------------------	-------------------------------------

## Returns

A reference to the newly-created hash map.

## 5.65.3.4 void ds\_map\_str\_insert ( ds\_map\_str map, ds\_str key, ds\_str value )

Inserts a key-value pair into a map.

The key and value are copied, so the caller may modify or `free()` them after calling this function.

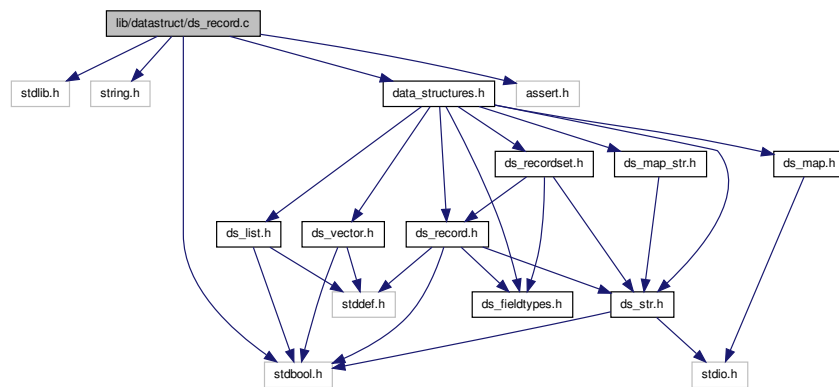
#### Parameters

<i>map</i>	A reference to the hash map.
<i>key</i>	The key.
<i>value</i>	The value.

## 5.66 lib/datastruct/ds\_record.c File Reference

Implementation of record database structure.

```
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_record.c:
```



### Data Structures

- struct [ds\\_record](#)

### Functions

- [ds\\_record ds\\_record\\_create](#) (const size\_t size)  
*Creates a new record.*
- void [ds\\_record\\_destroy](#) (ds\_record record)  
*Destroys a record and frees any associated resources.*
- void [ds\\_record\\_destructor](#) (void \*record)  
*A record destructor function.*
- void [ds\\_record\\_clear](#) (ds\_record record)  
*Clears and `free()`s all the elements in a record.*
- void [ds\\_record\\_set\\_field](#) (ds\_record record, const size\_t index, ds\_str field)  
*Sets a field of a record.*
- [ds\\_str ds\\_record\\_get\\_field](#) (ds\_record record, const size\_t index)  
*Retrieves the field at a specified index.*

- `size_t ds_record_size (ds_record record)`  
*Returns the size of a record.*
- `void ds_record_seek_start (ds_record record)`  
*Sets the current field to the first field of a record.*
- `ds_str ds_record_get_next_data (ds_record record)`  
*Returns the next field of the record.*
- `ds_record ds_record_tokenize (ds_str str, const char delim)`  
*Tokenizes a string into a record.*
- `ds_str ds_record_make_delim_string (ds_record record, const char delim)`  
*Makes a delimited string from a record.*
- `ds_str ds_record_make_values_string (ds_record record, enum ds_field_types *types)`  
*Makes a delimited SQL values string from a record.*

### 5.66.1 Detailed Description

Implementation of record database structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.66.2 Function Documentation

#### 5.66.2.1 `void ds_record_clear ( ds_record record )`

Clears and `free()`s all the elements in a record.

##### Parameters

<i>record</i>	The record.
---------------	-------------

#### 5.66.2.2 `ds_record ds_record_create ( const size_t size )`

Creates a new record.

##### Parameters

<i>size</i>	The size of the record.
-------------	-------------------------

##### Returns

A newly created record, or `NULL` on failure.

#### 5.66.2.3 `void ds_record_destroy ( ds_record record )`

Destroys a record and frees any associated resources.

## Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.66.2.4 `void ds_record_destructor ( void * record )`

A record destructor function.

## Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

5.66.2.5 `ds_str ds_record_get_field ( ds_record record, const size_t index )`

Retrieves the field at a specified index.

## Parameters

<i>record</i>	The record from which to retrieve.
<i>index</i>	The index of the desired field.

## Returns

A pointer to the field, or `NULL` if the index is out of range.

5.66.2.6 `ds_str ds_record_get_next_data ( ds_record record )`

Returns the next field of the record.

This function returns the data of the "current field", and advances the current field pointer. Subsequent calls to this function will return successive fields.

## Parameters

<i>record</i>	The record.
---------------	-------------

## Returns

A pointer to the next field, or `NULL` if the end of the record has been reached.

5.66.2.7 `ds_str ds_record_make_delim_string ( ds_record record, const char delim )`

Makes a delimited string from a record.

## Parameters

<i>record</i>	The record.
<i>delim</i>	The delimiting character.

## Returns

The delimited string, or `NULL` on failure.

5.66.2.8 `ds_str ds_record_make_values_string ( ds_record record, enum ds_field_types * types )`

Makes a delimited SQL values string from a record.

## Parameters

<i>record</i>	The record.
<i>types</i>	An array of types for each field, or <code>NULL</code> to assume they are all strings. The effect of this parameter is that string fields are quoted in the values string, whereas non-string fields are not.

## Returns

The delimited values string, or `NULL` on failure.

5.66.2.9 `void ds_record_seek_start ( ds_record record )`

Sets the current field to the first field of a record.

## Parameters

<i>record</i>	The record.
---------------	-------------

5.66.2.10 `void ds_record_set_field ( ds_record record, const size_t index, ds_str field )`

Sets a field of a record.

If the field is currently occupied, the existing field is `free()`d.

## Parameters

<i>record</i>	The record to set.
<i>index</i>	The index of the field to set.
<i>field</i>	The value to which to set the field.

5.66.2.11 `size_t ds_record_size ( ds_record record )`

Returns the size of a record.

## Parameters

<i>record</i>	The record.
---------------	-------------

## Returns

The size of the record.

5.66.2.12 `ds_record ds_record_tokenize ( ds_str str, const char delim )`

Tokenizes a string into a record.

## Parameters

<i>str</i>	The string to tokenize.
<i>delim</i>	The delimiting character.

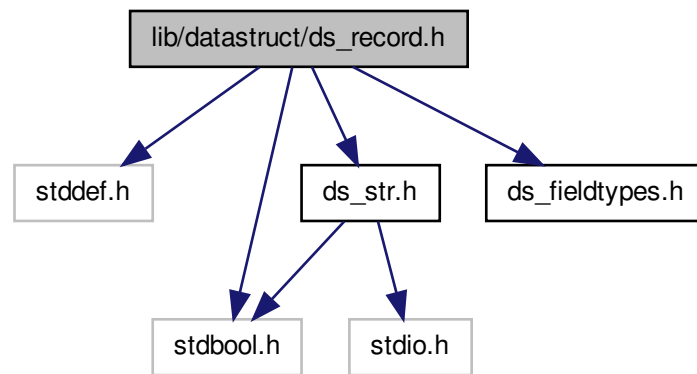
## Returns

A new record containing the tokens.

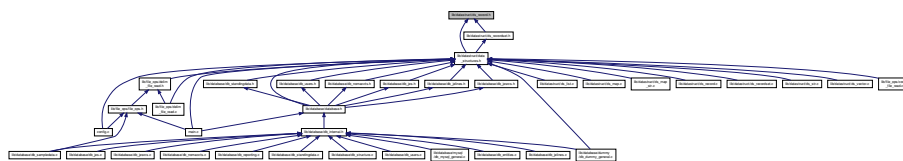
## 5.67 lib/datastruct/ds\_record.h File Reference

Interface to record data structure.

```
#include <stddef.h>
#include <stdbool.h>
#include "ds_str.h"
#include "ds_fieldtypes.h"
Include dependency graph for ds_record.h:
```



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef struct [ds\\_record](#) \* [ds\\_record](#)

## Functions

- [ds\\_record](#) [ds\\_record\\_create](#) (const [size\\_t](#) size)  
*Creates a new record.*
- void [ds\\_record\\_destroy](#) ([ds\\_record](#) record)  
*Destroys a record and frees any associated resources.*
- void [ds\\_record\\_destructor](#) (void \*record)

- A record destructor function.*
- void `ds_record_clear` (`ds_record` record)
- Clears and `free()`s all the elements in a record.*
- void `ds_record_set_field` (`ds_record` record, const `size_t` index, `ds_str` field)
- Sets a field of a record.*
- `ds_str` `ds_record_get_field` (`ds_record` record, const `size_t` index)
- Retrieves the field at a specified index.*
- `size_t` `ds_record_size` (`ds_record` record)
- Returns the size of a record.*
- void `ds_record_seek_start` (`ds_record` record)
- Sets the current field to the first field of a record.*
- `ds_str` `ds_record_get_next_data` (`ds_record` record)
- Returns the next field of the record.*
- `ds_record` `ds_record_tokenize` (`ds_str` str, const char delim)
- Tokenizes a string into a record.*
- `ds_str` `ds_record_make_delim_string` (`ds_record` record, const char delim)
- Makes a delimited string from a record.*
- `ds_str` `ds_record_make_values_string` (`ds_record` record, enum `ds_field_types` \*types)
- Makes a delimited SQL values string from a record.*

### 5.67.1 Detailed Description

Interface to record data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.67.2 Typedef Documentation

#### 5.67.2.1 typedef struct `ds_record*` `ds_record`

Typedef for opaque record datatype

### 5.67.3 Function Documentation

#### 5.67.3.1 void `ds_record_clear` ( `ds_record` record )

Clears and `free()`s all the elements in a record.

#### Parameters

<i>record</i>	The record.
---------------	-------------

#### 5.67.3.2 `ds_record` `ds_record_create` ( const `size_t` size )

Creates a new record.

## Parameters

<i>size</i>	The size of the record.
-------------	-------------------------

## Returns

A newly created record, or `NULL` on failure.

**5.67.3.3 void ds\_record\_destroy ( ds\_record record )**

Destroys a record and frees any associated resources.

## Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

**5.67.3.4 void ds\_record\_destructor ( void \* record )**

A record destructor function.

## Parameters

<i>record</i>	The record to destroy.
---------------	------------------------

**5.67.3.5 ds\_str ds\_record\_get\_field ( ds\_record record, const size\_t index )**

Retrieves the field at a specified index.

## Parameters

<i>record</i>	The record from which to retrieve.
<i>index</i>	The index of the desired field.

## Returns

A pointer to the field, or `NULL` if the index is out of range.

**5.67.3.6 ds\_str ds\_record\_get\_next\_data ( ds\_record record )**

Returns the next field of the record.

This function returns the data of the "current field", and advances the current field pointer. Subsequent calls to this function will return successive fields.

## Parameters

<i>record</i>	The record.
---------------	-------------

## Returns

A pointer to the next field, or `NULL` if the end of the record has been reached.



**5.67.3.7 ds\_str ds\_record\_make\_delim\_string ( ds\_record record, const char delim )**

Makes a delimited string from a record.

**Parameters**

<i>record</i>	The record.
<i>delim</i>	The delimiting character.

**Returns**

The delimited string, or `NULL` on failure.

**5.67.3.8 ds\_str ds\_record\_make\_values\_string ( ds\_record record, enum ds\_field\_types \* types )**

Makes a delimited SQL values string from a record.

**Parameters**

<i>record</i>	The record.
<i>types</i>	An array of types for each field, or <code>NULL</code> to assume they are all strings. The effect of this parameter is that string fields are quoted in the values string, whereas non-string fields are not.

**Returns**

The delimited values string, or `NULL` on failure.

**5.67.3.9 void ds\_record\_seek\_start ( ds\_record record )**

Sets the current field to the first field of a record.

**Parameters**

<i>record</i>	The record.
---------------	-------------

**5.67.3.10 void ds\_record\_set\_field ( ds\_record record, const size\_t index, ds\_str field )**

Sets a field of a record.

If the field is currently occupied, the existing field is `free()`d.

**Parameters**

<i>record</i>	The record to set.
<i>index</i>	The index of the field to set.
<i>field</i>	The value to which to set the field.

**5.67.3.11 size\_t ds\_record\_size ( ds\_record record )**

Returns the size of a record.

## Parameters

<i>record</i>	The record.
---------------	-------------

## Returns

The size of the record.

### 5.67.3.12 `ds_record ds_record_tokenize ( ds_str str, const char delim )`

Tokenizes a string into a record.

## Parameters

<i>str</i>	The string to tokenize.
<i>delim</i>	The delimiting character.

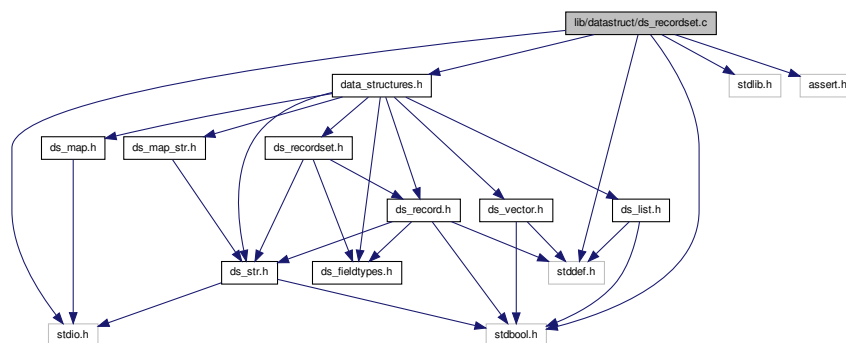
## Returns

A new record containing the tokens.

## 5.68 lib/datastruct/ds\_recordset.c File Reference

Implementation of query result set structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_recordset.c:
```



## Data Structures

- struct [ds\\_recordset](#)

## Functions

- [ds\\_recordset ds\\_recordset\\_create](#) (const size\_t num\_fields)

- Creates a new record set.*
- void `ds_recordset_destroy` (`ds_recordset` set)
- Destroys a record set and frees associated resources.*
- `ds_record` `ds_recordset_add_record` (`ds_recordset` set, `ds_record` record)
- Adds a record to a record set.*
- size\_t `ds_recordset_num_fields` (`ds_recordset` set)
- Returns the number of fields in a record set.*
- size\_t `ds_recordset_num_records` (`ds_recordset` set)
- Returns the number of records in a record set.*
- void `ds_recordset_set_headers` (`ds_recordset` set, `ds_record` headers)
- Sets the record headers in a record set.*
- void `ds_recordset_set_type` (`ds_recordset` set, const size\_t index, const enum `ds_field_types` type)
- Sets the type for a specified field.*
- `ds_str` `ds_recordset_get_text_report` (`ds_recordset` set)
- Returns a formatted text report for the record set.*
- void `ds_recordset_seek_start` (`ds_recordset` set)
- Sets the current record to the first record.*
- `ds_record` `ds_recordset_next_record` (`ds_recordset` set)
- Returns the next record in the record set.*
- `ds_str` `ds_recordset_get_next_insert_query` (`ds_recordset` set, const char \*table\_name)
- Gets the next SQL INSERT query.*

### 5.68.1 Detailed Description

Implementation of query result set structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.68.2 Function Documentation

#### 5.68.2.1 `ds_record` `ds_recordset_add_record` ( `ds_recordset` set, `ds_record` record )

Adds a record to a record set.

The record *must* have the same number of fields as the number of fields provided to `ds_recordset_create()`.

#### Parameters

<i>set</i>	The record set to which to add.
<i>record</i>	The record to add.

#### Returns

A pointer to the new record (i.e. it returns the second parameter) or `NULL` on failure.

### 5.68.2.2 `ds_recordset ds_recordset_create ( const size_t num_fields )`

Creates a new record set.

#### Parameters

<i>num_fields</i>	The non-zero number of fields in the record set.
-------------------	--

#### Returns

A pointer to the new record set.

### 5.68.2.3 `void ds_recordset_destroy ( ds_recordset set )`

Destroys a record set and frees associated resources.

#### Parameters

<i>set</i>	The record set to destroy.
------------	----------------------------

### 5.68.2.4 `ds_str ds_recordset_get_next_insert_query ( ds_recordset set, const char * table_name )`

Gets the next SQL INSERT query.

#### Parameters

<i>set</i>	The set.
<i>table_name</i>	The table name into which to insert.

#### Returns

The query. Caller is responsible for `free()` ing.

### 5.68.2.5 `ds_str ds_recordset_get_text_report ( ds_recordset set )`

Returns a formatted text report for the record set.

The report is returned as a single multi-line string.

#### Parameters

<i>set</i>	The record set.
------------	-----------------

#### Returns

A pointer to the report. The caller is responsible for `free()` ing this pointer.

### 5.68.2.6 `ds_record ds_recordset_next_record ( ds_recordset set )`

Returns the next record in the record set.

This function returns the "current record", and advances the current record pointer. Subsequent calls to this function will return successive records.

## Parameters

<i>set</i>	The record set.
------------	-----------------

## Returns

A pointer to the next record, or `NULL` if the end of the record set has been reached.

5.68.2.7 `size_t ds_recordset_num_fields ( ds_recordset set )`

Returns the number of fields in a record set.

## Parameters

<i>set</i>	The record set.
------------	-----------------

## Returns

The number of fields in the record set.

5.68.2.8 `size_t ds_recordset_num_records ( ds_recordset set )`

Returns the number of records in a record set.

## Parameters

<i>set</i>	The record set.
------------	-----------------

## Returns

The number of records in the record set.

5.68.2.9 `void ds_recordset_seek_start ( ds_recordset set )`

Sets the current record to the first record.

## Parameters

<i>set</i>	The record set.
------------	-----------------

5.68.2.10 `void ds_recordset_set_headers ( ds_recordset set, ds_record headers )`

Sets the record headers in a record set.

## Parameters

<i>set</i>	The record set.
<i>headers</i>	The headers, in the form of a <a href="#">ds_record</a> of strings. The list <i>must</i> have the same number of elements as the number of fields provided to <a href="#">ds_recordset_create()</a> .

5.68.2.11 `void ds_recordset_set_type ( ds_recordset set, const size_t index, const enum ds_field_types type )`

Sets the type for a specified field.

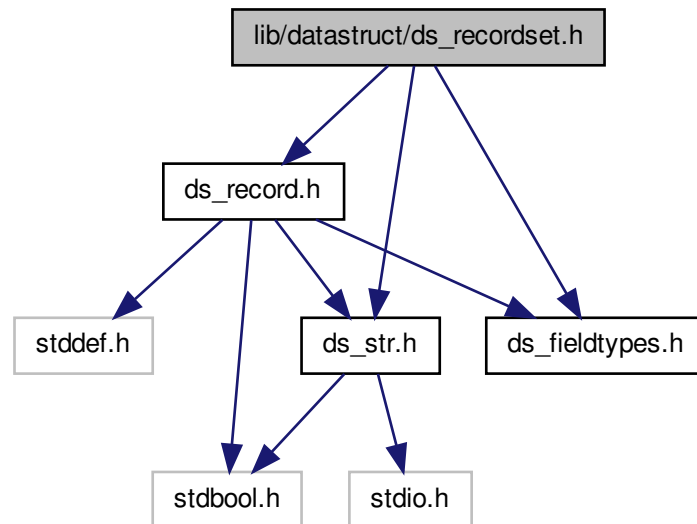
## Parameters

<i>set</i>	The record set.
<i>index</i>	The index to set.
<i>type</i>	The type for the field at the specified index.

## 5.69 lib/datastruct/ds\_recordset.h File Reference

Interface to record set structure.

```
#include "ds_record.h"
#include "ds_str.h"
#include "ds_fieldtypes.h"
Include dependency graph for ds_recordset.h:
```



This graph shows which files directly or indirectly include this file:



### Typedefs

- typedef struct [ds\\_recordset](#) \* [ds\\_recordset](#)

### Functions

- [ds\\_recordset ds\\_recordset\\_create](#) (const size\_t num\_fields)

- Creates a new record set.*
- void `ds_recordset_destroy` (`ds_recordset` set)
- Destroys a record set and frees associated resources.*
- `ds_record` `ds_recordset_add_record` (`ds_recordset` set, `ds_record` record)
- Adds a record to a record set.*
- size\_t `ds_recordset_num_fields` (`ds_recordset` set)
- Returns the number of fields in a record set.*
- size\_t `ds_recordset_num_records` (`ds_recordset` set)
- Returns the number of records in a record set.*
- void `ds_recordset_set_headers` (`ds_recordset` set, `ds_record` headers)
- Sets the record headers in a record set.*
- void `ds_recordset_set_type` (`ds_recordset` set, const size\_t index, const enum `ds_field_types` type)
- Sets the type for a specified field.*
- `ds_str` `ds_recordset_get_text_report` (`ds_recordset` set)
- Returns a formatted text report for the record set.*
- `ds_str` `ds_recordset_get_next_insert_query` (`ds_recordset` set, const char \*table\_name)
- Gets the next SQL INSERT query.*
- void `ds_recordset_seek_start` (`ds_recordset` set)
- Sets the current record to the first record.*
- `ds_record` `ds_recordset_next_record` (`ds_recordset` set)
- Returns the next record in the record set.*

### 5.69.1 Detailed Description

Interface to record set structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.69.2 Typedef Documentation

#### 5.69.2.1 typedef struct `ds_recordset*` `ds_recordset`

Typedef for opaque record set data type

### 5.69.3 Function Documentation

#### 5.69.3.1 `ds_record` `ds_recordset_add_record` ( `ds_recordset` set, `ds_record` record )

Adds a record to a record set.

The record *must* have the same number of fields as the number of fields provided to `ds_recordset_create()`.

#### Parameters

<i>set</i>	The record set to which to add.
<i>record</i>	The record to add.

**Returns**

A pointer to the new record (i.e. it returns the second parameter) or `NULL` on failure.

**5.69.3.2 `ds_recordset ds_recordset_create ( const size_t num_fields )`**

Creates a new record set.

**Parameters**

<i>num_fields</i>	The non-zero number of fields in the record set.
-------------------	--

**Returns**

A pointer to the new record set.

**5.69.3.3 `void ds_recordset_destroy ( ds_recordset set )`**

Destroys a record set and frees associated resources.

**Parameters**

<i>set</i>	The record set to destroy.
------------	----------------------------

**5.69.3.4 `ds_str ds_recordset_get_next_insert_query ( ds_recordset set, const char * table_name )`**

Gets the next SQL INSERT query.

**Parameters**

<i>set</i>	The set.
<i>table_name</i>	The table name into which to insert.

**Returns**

The query. Caller is responsible for `free()` ing.

**5.69.3.5 `ds_str ds_recordset_get_text_report ( ds_recordset set )`**

Returns a formatted text report for the record set.

The report is returned as a single multi-line string.

**Parameters**

<i>set</i>	The record set.
------------	-----------------

**Returns**

A pointer to the report. The caller is responsible for `free()` ing this pointer.

**5.69.3.6 `ds_record ds_recordset_next_record ( ds_recordset set )`**

Returns the next record in the record set.



This function returns the "current record", and advances the current record pointer. Subsequent calls to this function will return successive records.

#### Parameters

<i>set</i>	The record set.
------------	-----------------

#### Returns

A pointer to the next record, or `NULL` if the end of the record set has been reached.

#### 5.69.3.7 `size_t ds_recordset_num_fields ( ds_recordset set )`

Returns the number of fields in a record set.

#### Parameters

<i>set</i>	The record set.
------------	-----------------

#### Returns

The number of fields in the record set.

#### 5.69.3.8 `size_t ds_recordset_num_records ( ds_recordset set )`

Returns the number of records in a record set.

#### Parameters

<i>set</i>	The record set.
------------	-----------------

#### Returns

The number of records in the record set.

#### 5.69.3.9 `void ds_recordset_seek_start ( ds_recordset set )`

Sets the current record to the first record.

#### Parameters

<i>set</i>	The record set.
------------	-----------------

#### 5.69.3.10 `void ds_recordset_set_headers ( ds_recordset set, ds_record headers )`

Sets the record headers in a record set.

#### Parameters

<i>set</i>	The record set.
<i>headers</i>	The headers, in the form of a <a href="#">ds_record</a> of strings. The list <i>must</i> have the same number of elements as the number of fields provided to <a href="#">ds_recordset_create()</a> .

5.69.3.11 `void ds_recordset_set_type ( ds_recordset set, const size_t index, const enum ds_field_types type )`

Sets the type for a specified field.

#### Parameters

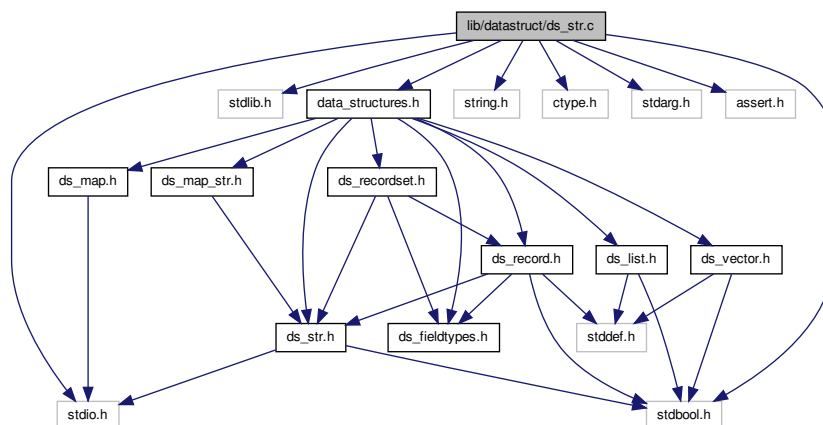
<code>set</code>	The record set.
<code>index</code>	The index to set.
<code>type</code>	The type for the field at the specified index.

## 5.70 lib/datastruct/ds\_str.c File Reference

Implementation of string data structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
#include <stdarg.h>
#include <assert.h>
#include "data_structures.h"
```

Include dependency graph for ds\_str.c:



## Data Structures

- struct [ds\\_str](#)

## Functions

- [ds\\_str ds\\_str\\_create\\_direct](#) (char \*init\_str, const size\_t init\_str\_size)  
*Creates a string using allocated memory.*
- [ds\\_str ds\\_str\\_create](#) (const char \*init\_str)  
*Creates a new string from a C-style string.*
- [ds\\_str ds\\_str\\_dup](#) (ds\_str src)  
*Creates a new string from another string.*

- [ds\\_str ds\\_str\\_create\\_sprintf](#) (const char \*format,...)  
*Creates a string with `sprintf()`-type format.*
- void [ds\\_str\\_destroy](#) (ds\_str str)  
*Destroys a string and releases allocated resources.*
- void [ds\\_str\\_destructor](#) (void \*str)  
*Destroys a string and releases allocated resources.*
- [ds\\_str ds\\_str\\_assign](#) (ds\_str dst, ds\_str src)  
*Assigns a string to another.*
- [ds\\_str ds\\_str\\_assign\\_cstr](#) (ds\_str dst, const char \*src)  
*Assigns a C-style string to a string.*
- const char \* [ds\\_str\\_cstr](#) (ds\_str str)  
*Returns a C-style string containing the string's contents.*
- size\_t [ds\\_str\\_length](#) (ds\_str str)  
*Returns the length of a string.*
- [ds\\_str ds\\_str\\_size\\_to\\_fit](#) (ds\_str str)  
*Reduces a string's capacity to fit its length.*
- [ds\\_str ds\\_str\\_concat](#) (ds\_str dst, ds\_str src)  
*Concatenates two strings.*
- [ds\\_str ds\\_str\\_concat\\_cstr](#) (ds\_str dst, const char \*src)  
*Concatenates a C-style string to a string.*
- [ds\\_str ds\\_str\\_trunc](#) (ds\_str str, const size\_t length)  
*Truncates a string.*
- unsigned long [ds\\_str\\_hash](#) (ds\_str str)  
*Calculates a hash of a string.*
- int [ds\\_str\\_compare](#) (ds\_str s1, ds\_str s2)  
*Compares two strings.*
- int [ds\\_str\\_compare\\_cstr](#) (ds\_str s1, const char \*s2)  
*Compares a string with a C-style string.*
- int [ds\\_str\\_strchr](#) (ds\_str str, const char ch, const int start)  
*Returns index of first occurrence of a character.*
- [ds\\_str ds\\_str\\_substr\\_left](#) (ds\_str str, const size\_t numchars)  
*Returns a left substring.*
- [ds\\_str ds\\_str\\_substr\\_right](#) (ds\_str str, const size\_t numchars)  
*Returns a right substring.*
- void [ds\\_str\\_split](#) (ds\_str src, ds\_str \*left, ds\_str \*right, const char sc)  
*Splits a string.*
- void [ds\\_str\\_trim\\_leading](#) (ds\_str str)  
*Trims leading whitespace in-place.*
- void [ds\\_str\\_trim\\_trailing](#) (ds\_str str)  
*Trims trailing whitespace in-place.*
- void [ds\\_str\\_trim](#) (ds\_str str)  
*Trims leading and trailing whitespace in-place.*
- char [ds\\_str\\_char\\_at\\_index](#) (ds\_str str, const size\_t index)  
*Returns the character at a specified index.*
- bool [ds\\_str\\_is\\_empty](#) (ds\_str str)  
*Checks if a string is empty.*
- bool [ds\\_str\\_is\\_alnum](#) (ds\_str str)  
*Checks if a string contains only alphanumeric characters.*
- void [ds\\_str\\_clear](#) (ds\_str str)  
*Clears (empties) a string.*
- bool [ds\\_str\\_intval](#) (ds\_str str, const int base, int \*value)

*Gets the integer value of a string.*

- bool `ds_str_doubleval` (`ds_str` str, double \*value)

*Gets the double value of a string.*

- `ds_str ds_str_getline` (`ds_str` str, const size\_t size, FILE \*fp)

*Gets a line from a file and assigns it to a string.*

- `ds_str ds_str_decorate` (`ds_str` str, `ds_str` left\_dec, `ds_str` right\_dec)

*Brackets a string with decoration strings.*

### 5.70.1 Detailed Description

Implementation of string data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.70.2 Function Documentation

#### 5.70.2.1 `ds_str ds_str_assign ( ds_str dst, ds_str src )`

Assigns a string to another.

##### Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source string.

##### Returns

*dst* on success, NULL on failure.

#### 5.70.2.2 `ds_str ds_str_assign_cstr ( ds_str dst, const char * src )`

Assigns a C-style string to a string.

##### Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source C-style string.

##### Returns

*dst* on success, NULL on failure.

#### 5.70.2.3 `char ds_str_char_at_index ( ds_str str, const size_t index )`

Returns the character at a specified index.

## Parameters

<i>str</i>	The string.
<i>index</i>	The specified index.

## Returns

The character at the specified index.

**5.70.2.4 void ds\_str\_clear ( ds\_str str )**

Clears (empties) a string.

## Parameters

<i>str</i>	The string.
------------	-------------

**5.70.2.5 int ds\_str\_compare ( ds\_str s1, ds\_str s2 )**

Compares two strings.

## Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second string.

## Returns

Less than, equal to, or greater than zero if s1 is found, respectively, to be less than, equal to, or greater than s2.

**5.70.2.6 int ds\_str\_compare\_cstr ( ds\_str s1, const char \* s2 )**

Compares a string with a C-style string.

## Parameters

<i>s1</i>	The first string.
<i>s2</i>	The second, C-Style string.

## Returns

Less than, equal to, or greater than zero if s1 is found, respectively, to be less than, equal to, or greater than s2.

**5.70.2.7 ds\_str ds\_str\_concat ( ds\_str dst, ds\_str src )**

Concatenates two strings.

## Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

**Returns**

The destination string, or `NULL` on failure.

**5.70.2.8 `ds_str ds_str_concat_cstr ( ds_str dst, const char * src )`**

Concatenates a C-style string to a string.

**Parameters**

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

**Returns**

The destination string, or `NULL` on failure.

**5.70.2.9 `ds_str ds_str_create ( const char * init_str )`**

Creates a new string from a C-style string.

**Parameters**

<i>init_str</i>	The C-style string.
-----------------	---------------------

**Returns**

The new string, or `NULL` on failure.

**5.70.2.10 `ds_str ds_str_create_direct ( char * init_str, const size_t init_str_size )`**

Creates a string using allocated memory.

The normal construction functions duplicate the string used to create it. In cases where allocated memory is already available (e.g. in `ds_str_create_sprintf()`) this function allows that memory to be directly assigned to the string, avoiding an unnecessary duplication.

**Parameters**

<i>init_str</i>	The allocated memory. IMPORTANT: If the construction of the string fails, this memory will be <code>free()</code> d.
<i>init_str_size</i>	The size of the allocated memory. IMPORTANT: The string's length is assumed to be one less than this quantity, and a call to <code>strlen()</code> is NOT performed.

**Returns**

The new string, or `NULL` on failure.

**5.70.2.11 `ds_str ds_str_create_sprintf ( const char * format, ... )`**

Creates a string with `sprintf()`-type format.

## Parameters

<i>format</i>	The format string.
...	The subsequent arguments as specified by the format string.

## Returns

The new string, or `NULL` on failure.

5.70.2.12 `const char* ds_str_cstr ( ds_str str )`

Returns a C-style string containing the string's contents.

## Parameters

<i>str</i>	The string.
------------	-------------

## Returns

The C-style string containing the string's contents. The caller should not directly modify this string.

5.70.2.13 `ds_str ds_str_decorate ( ds_str str, ds_str left_dec, ds_str right_dec )`

Brackets a string with decoration strings.

## Parameters

<i>str</i>	The string to decorate.
<i>left_dec</i>	The string to add to the left of <i>str</i> .
<i>right_dec</i>	The string to add to the right of <i>str</i> , or <code>NULL</code> to add <i>left_dec</i> to both sides.

## Returns

The decorated string.

5.70.2.14 `void ds_str_destroy ( ds_str str )`

Destroys a string and releases allocated resources.

## Parameters

<i>str</i>	The string to destroy..
------------	-------------------------

5.70.2.15 `void ds_str_destructor ( void * str )`

Destroys a string and releases allocated resources.

This function calls `ds_str_destroy()`, and can be passed to a data structure expecting a destructor function with the signature `void (*)(void *)`.

## Parameters

<i>str</i>	The string to destroy.
------------	------------------------

**5.70.2.16** `bool ds_str_doubleval ( ds_str str, double * value )`

Gets the double value of a string.

**Parameters**

<i>str</i>	The string.
<i>value</i>	A pointer to the double in which to store the value. Zero is stored if the string does not contain a valid double value.

**Returns**

`true` on successful conversion, `false` if the string does not contain a valid double value.

**5.70.2.17** `ds_str ds_str_dup ( ds_str src )`

Creates a new string from another string.

**Parameters**

<i>src</i>	The other string.
------------	-------------------

**Returns**

The new string, or `NULL` on failure.

**5.70.2.18** `ds_str ds_str_getline ( ds_str str, const size_t size, FILE * fp )`

Gets a line from a file and assigns it to a string.

Any trailing newline character is stripped.

**Parameters**

<i>str</i>	The string.
<i>size</i>	The maximum number of bytes to read, including the null.
<i>fp</i>	The file pointer from which to read.

**Returns**

`dst`

**5.70.2.19** `unsigned long ds_str_hash ( ds_str str )`

Calculates a hash of a string.

Uses Dan Bernstein's djb2 algorithm.

**Parameters**

<i>str</i>	The string.
------------	-------------

**Returns**

The hash value



**5.70.2.20** `bool ds_str_intval ( ds_str str, const int base, int * value )`

Gets the integer value of a string.

**Parameters**

<i>str</i>	The string.
<i>base</i>	The base of the integer. This has the same meaning as the third argument to standard C <code>strtol()</code> .
<i>value</i>	A pointer to the integer in which to store the value. Zero is stored if the string does not contain a valid integer value.

**Returns**

`true` on successful conversion, `false` if the string does not contain a valid integer value.

**5.70.2.21** `bool ds_str_is_alnum ( ds_str str )`

Checks if a string contains only alphanumeric characters.

The string must contain *some* alphanumeric characters to check `true`, i.e. the string must be non-empty. Thus it can be used to check that a string does indeed contain content, and that that content is solely alphanumeric.

**Parameters**

<i>str</i>	The string.
------------	-------------

**Returns**

`true` if the string contains only alphanumeric characters, `false` otherwise.

**5.70.2.22** `bool ds_str_is_empty ( ds_str str )`

Checks if a string is empty.

**Parameters**

<i>str</i>	The string.
------------	-------------

**Returns**

`true` if the string is empty, `false` otherwise.

**5.70.2.23** `size_t ds_str_length ( ds_str str )`

Returns the length of a string.

**Parameters**

<i>str</i>	The string.
------------	-------------

**Returns**

The length of the string.

**5.70.2.24 ds\_str ds\_str\_size\_to\_fit ( ds\_str str )**

Reduces a string's capacity to fit its length.

**Parameters**

<i>str</i>	The string to size.
------------	---------------------

**Returns**

*str*, or NULL on failure.

**5.70.2.25 void ds\_str\_split ( ds\_str src, ds\_str \* left, ds\_str \* right, const char sc )**

Splits a string.

**Parameters**

<i>src</i>	The string to split.
<i>left</i>	Pointer to left substring (modified)
<i>right</i>	Pointer to right substring (modified)
<i>sc</i>	Split character.

**5.70.2.26 int ds\_str\_strchr ( ds\_str str, const char ch, const int start )**

Returns index of first occurrence of a character.

**Parameters**

<i>str</i>	The string.
<i>ch</i>	The character for which to search.
<i>start</i>	The index of the string at which to start looking. Set this to non-zero to begin searching from a point other than the first character of the string.

**Returns**

The index of the first occurrence, or -1 if the character was not found.

**5.70.2.27 ds\_str ds\_str\_substr\_left ( ds\_str str, const size\_t numchars )**

Returns a left substring.

**Parameters**

<i>str</i>	The string.
<i>numchars</i>	The number of left characters to return. If this is greater than the length of the string, the whole string is returned.

**Returns**

A new string representing the substring.

**5.70.2.28 ds\_str ds\_str\_substr\_right ( ds\_str str, const size\_t numchars )**

Returns a right substring.

**Parameters**

<i>str</i>	The string.
<i>numchars</i>	The number of right characters to return. If this is greater than the length of the string, the whole string is returned.

**Returns**

A new string representing the substring.

**5.70.2.29 void ds\_str\_trim ( ds\_str str )**

Trims leading and trailing whitespace in-place.

**Parameters**

<i>str</i>	The string.
------------	-------------

**5.70.2.30 void ds\_str\_trim\_leading ( ds\_str str )**

Trims leading whitespace in-place.

**Parameters**

<i>str</i>	The string.
------------	-------------

**5.70.2.31 void ds\_str\_trim\_trailing ( ds\_str str )**

Trims trailing whitespace in-place.

**Parameters**

<i>str</i>	The string.
------------	-------------

**5.70.2.32 ds\_str ds\_str\_trunc ( ds\_str str, const size\_t length )**

Truncates a string.

**Parameters**

<i>str</i>	The string.
<i>length</i>	The new length to which to truncate.

## Returns

The original string, or `NULL` on failure.

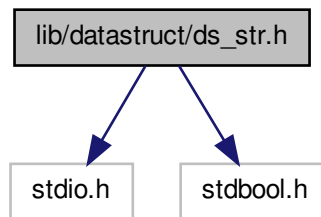
## 5.71 lib/datastruct/ds\_str.h File Reference

Interface to string data structure.

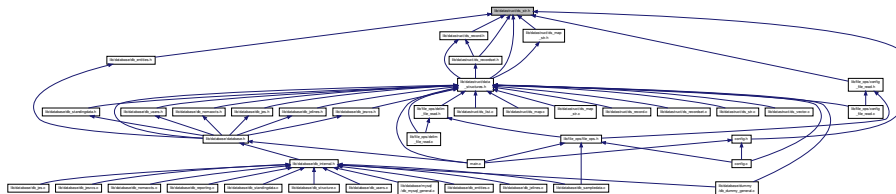
```
#include <stdio.h>
```

```
#include <stdbool.h>
```

Include dependency graph for `ds_str.h`:



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef struct [ds\\_str](#) \* [ds\\_str](#)

## Functions

- [ds\\_str ds\\_str\\_create](#) (const char \*init\_str)  
*Creates a new string from a C-style string.*
- [ds\\_str ds\\_str\\_dup](#) (ds\_str src)  
*Creates a new string from another string.*
- [ds\\_str ds\\_str\\_create\\_sprintf](#) (const char \*format,...)  
*Creates a string with `sprintf()`-type format.*
- [ds\\_str ds\\_str\\_create\\_direct](#) (char \*init\_str, const size\_t init\_str\_size)  
*Creates a string using allocated memory.*
- void [ds\\_str\\_destroy](#) (ds\_str str)  
*Destroys a string and releases allocated resources.*

- void [ds\\_str\\_destructor](#) (void \*str)  
*Destroys a string and releases allocated resources.*
- [ds\\_str ds\\_str\\_assign](#) (ds\_str dst, ds\_str src)  
*Assigns a string to another.*
- [ds\\_str ds\\_str\\_assign\\_cstr](#) (ds\_str dst, const char \*src)  
*Assigns a C-style string to a string.*
- const char \* [ds\\_str\\_cstr](#) (ds\_str str)  
*Returns a C-style string containing the string's contents.*
- size\_t [ds\\_str\\_length](#) (ds\_str str)  
*Returns the length of a string.*
- [ds\\_str ds\\_str\\_size\\_to\\_fit](#) (ds\_str str)  
*Reduces a string's capacity to fit its length.*
- [ds\\_str ds\\_str\\_concat](#) (ds\_str dst, ds\_str src)  
*Concatenates two strings.*
- [ds\\_str ds\\_str\\_concat\\_cstr](#) (ds\_str dst, const char \*src)  
*Concatenates a C-style string to a string.*
- [ds\\_str ds\\_str\\_trunc](#) (ds\_str str, const size\_t length)  
*Truncates a string.*
- unsigned long [ds\\_str\\_hash](#) (ds\_str str)  
*Calculates a hash of a string.*
- int [ds\\_str\\_compare](#) (ds\_str s1, ds\_str s2)  
*Compares two strings.*
- int [ds\\_str\\_compare\\_cstr](#) (ds\_str s1, const char \*s2)  
*Compares a string with a C-style string.*
- int [ds\\_str\\_strchr](#) (ds\_str str, const char ch, const int start)  
*Returns index of first occurrence of a character.*
- [ds\\_str ds\\_str\\_substr\\_left](#) (ds\_str str, const size\_t numchars)  
*Returns a left substring.*
- [ds\\_str ds\\_str\\_substr\\_right](#) (ds\_str str, const size\_t numchars)  
*Returns a right substring.*
- void [ds\\_str\\_split](#) (ds\_str src, ds\_str \*left, ds\_str \*right, const char sc)  
*Splits a string.*
- void [ds\\_str\\_trim\\_leading](#) (ds\_str str)  
*Trims leading whitespace in-place.*
- void [ds\\_str\\_trim\\_trailing](#) (ds\_str str)  
*Trims trailing whitespace in-place.*
- void [ds\\_str\\_trim](#) (ds\_str str)  
*Trims leading and trailing whitespace in-place.*
- char [ds\\_str\\_char\\_at\\_index](#) (ds\_str str, const size\_t index)  
*Returns the character at a specified index.*
- bool [ds\\_str\\_is\\_empty](#) (ds\_str str)  
*Checks if a string is empty.*
- bool [ds\\_str\\_is\\_alnum](#) (ds\_str str)  
*Checks if a string contains only alphanumeric characters.*
- void [ds\\_str\\_clear](#) (ds\_str str)  
*Clears (empties) a string.*
- bool [ds\\_str\\_intval](#) (ds\_str str, const int base, int \*value)  
*Gets the integer value of a string.*
- bool [ds\\_str\\_doubleval](#) (ds\_str str, double \*value)  
*Gets the double value of a string.*
- [ds\\_str ds\\_str\\_getline](#) (ds\_str str, const size\_t size, FILE \*fp)  
*Gets a line from a file and assigns it to a string.*
- [ds\\_str ds\\_str\\_decorate](#) (ds\_str str, ds\_str left\_dec, ds\_str right\_dec)  
*Brackets a string with decoration strings.*

### 5.71.1 Detailed Description

Interface to string data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.71.2 Typedef Documentation

#### 5.71.2.1 typedef struct **ds\_str**\* **ds\_str**

Opaque data type for string

### 5.71.3 Function Documentation

#### 5.71.3.1 **ds\_str** **ds\_str\_assign** ( **ds\_str** *dst*, **ds\_str** *src* )

Assigns a string to another.

#### Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source string.

#### Returns

*dst* on success, `NULL` on failure.

#### 5.71.3.2 **ds\_str** **ds\_str\_assign\_cstr** ( **ds\_str** *dst*, `const char *` *src* )

Assigns a C-style string to a string.

#### Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source C-style string.

#### Returns

*dst* on success, `NULL` on failure.

#### 5.71.3.3 `char` **ds\_str\_char\_at\_index** ( **ds\_str** *str*, `const size_t` *index* )

Returns the character at a specified index.

#### Parameters

<i>str</i>	The string.
<i>index</i>	The specified index.

**Returns**

The character at the specified index.

**5.71.3.4 void ds\_str\_clear ( ds\_str str )**

Clears (empties) a string.

**Parameters**

<i>str</i>	The string.
------------	-------------

**5.71.3.5 int ds\_str\_compare ( ds\_str s1, ds\_str s2 )**

Compares two strings.

**Parameters**

<i>s1</i>	The first string.
<i>s2</i>	The second string.

**Returns**

Less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, equal to, or greater than *s2*.

**5.71.3.6 int ds\_str\_compare\_cstr ( ds\_str s1, const char \* s2 )**

Compares a string with a C-style string.

**Parameters**

<i>s1</i>	The first string.
<i>s2</i>	The second, C-Style string.

**Returns**

Less than, equal to, or greater than zero if *s1* is found, respectively, to be less than, equal to, or greater than *s2*.

**5.71.3.7 ds\_str ds\_str\_concat ( ds\_str dst, ds\_str src )**

Concatenates two strings.

**Parameters**

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

**Returns**

The destination string, or `NULL` on failure.

#### 5.71.3.8 `ds_str ds_str_concat_cstr ( ds_str dst, const char * src )`

Concatenates a C-style string to a string.

##### Parameters

<i>dst</i>	The destination string.
<i>src</i>	The source strings.

##### Returns

The destination string, or `NULL` on failure.

#### 5.71.3.9 `ds_str ds_str_create ( const char * init_str )`

Creates a new string from a C-style string.

##### Parameters

<i>init_str</i>	The C-style string.
-----------------	---------------------

##### Returns

The new string, or `NULL` on failure.

#### 5.71.3.10 `ds_str ds_str_create_direct ( char * init_str, const size_t init_str_size )`

Creates a string using allocated memory.

The normal construction functions duplicate the string used to create it. In cases where allocated memory is already available (e.g. in `ds_str_create_sprintf()`) this function allows that memory to be directly assigned to the string, avoiding an unnecessary duplication.

##### Parameters

<i>init_str</i>	The allocated memory. IMPORTANT: If the construction of the string fails, this memory will be <code>free()</code> d.
<i>init_str_size</i>	The size of the allocated memory. IMPORTANT: The string's length is assumed to be one less than this quantity, and a call to <code>strlen()</code> is NOT performed.

##### Returns

The new string, or `NULL` on failure.

#### 5.71.3.11 `ds_str ds_str_create_sprintf ( const char * format, ... )`

Creates a string with `sprintf()`-type format.

##### Parameters

<i>format</i>	The format string.
<i>...</i>	The subsequent arguments as specified by the format string.



**Returns**

The new string, or `NULL` on failure.

**5.71.3.12 `const char* ds_str_cstr ( ds_str str )`**

Returns a C-style string containing the string's contents.

**Parameters**

<i>str</i>	The string.
------------	-------------

**Returns**

The C-style string containing the string's contents. The caller should not directly modify this string.

**5.71.3.13 `ds_str ds_str_decorate ( ds_str str, ds_str left_dec, ds_str right_dec )`**

Brackets a string with decoration strings.

**Parameters**

<i>str</i>	The string to decorate.
<i>left_dec</i>	The string to add to the left of <i>str</i> .
<i>right_dec</i>	The string to add to the right of <i>str</i> , or <code>NULL</code> to add <i>left_dec</i> to both sides.

**Returns**

The decorated string.

**5.71.3.14 `void ds_str_destroy ( ds_str str )`**

Destroys a string and releases allocated resources.

**Parameters**

<i>str</i>	The string to destroy..
------------	-------------------------

**5.71.3.15 `void ds_str_destructor ( void * str )`**

Destroys a string and releases allocated resources.

This function calls `ds_str_destroy()`, and can be passed to a data structure expecting a destructor function with the signature `void (*)(void *)`.

**Parameters**

<i>str</i>	The string to destroy.
------------	------------------------

**5.71.3.16 `bool ds_str_doubleval ( ds_str str, double * value )`**

Gets the double value of a string.

## Parameters

<i>str</i>	The string.
<i>value</i>	A pointer to the double in which to store the value. Zero is stored if the string does not contain a valid double value.

## Returns

`true` on successful conversion, `false` if the string does not contain a valid double value.

**5.71.3.17 ds\_str ds\_str\_dup ( ds\_str src )**

Creates a new string from another string.

## Parameters

<i>src</i>	The other string.
------------	-------------------

## Returns

The new string, or `NULL` on failure.

**5.71.3.18 ds\_str ds\_str\_getline ( ds\_str str, const size\_t size, FILE \* fp )**

Gets a line from a file and assigns it to a string.

Any trailing newline character is stripped.

## Parameters

<i>str</i>	The string.
<i>size</i>	The maximum number of bytes to read, including the null.
<i>fp</i>	The file pointer from which to read.

## Returns

`dst`

**5.71.3.19 unsigned long ds\_str\_hash ( ds\_str str )**

Calculates a hash of a string.

Uses Dan Bernstein's djb2 algorithm.

## Parameters

<i>str</i>	The string.
------------	-------------

## Returns

The hash value

**5.71.3.20 bool ds\_str\_intval ( ds\_str str, const int base, int \* value )**

Gets the integer value of a string.

## Parameters

<i>str</i>	The string.
<i>base</i>	The base of the integer. This has the same meaning as the third argument to standard C <code>strtol()</code> .
<i>value</i>	A pointer to the integer in which to store the value. Zero is stored if the string does not contain a valid integer value.

## Returns

`true` on successful conversion, `false` if the string does not contain a valid integer value.

5.71.3.21 `bool ds_str_is_alnum ( ds_str str )`

Checks if a string contains only alphanumeric characters.

The string must contain *some* alphanumeric characters to check `true`, i.e. the string must be non-empty. Thus it can be used to check that a string does indeed contain content, and that that content is solely alphanumeric.

## Parameters

<i>str</i>	The string.
------------	-------------

## Returns

`true` if the string contains only alphanumeric characters, `false` otherwise.

5.71.3.22 `bool ds_str_is_empty ( ds_str str )`

Checks if a string is empty.

## Parameters

<i>str</i>	The string.
------------	-------------

## Returns

`true` if the string is empty, `false` otherwise.

5.71.3.23 `size_t ds_str_length ( ds_str str )`

Returns the length of a string.

## Parameters

<i>str</i>	The string.
------------	-------------

## Returns

The length of the string.

5.71.3.24 `ds_str ds_str_size_to_fit ( ds_str str )`

Reduces a string's capacity to fit its length.

## Parameters

<i>str</i>	The string to size.
------------	---------------------

## Returns

*str*, or NULL on failure.

**5.71.3.25** void *ds\_str\_split* ( *ds\_str src*, *ds\_str \* left*, *ds\_str \* right*, const char *sc* )

Splits a string.

## Parameters

<i>src</i>	The string to split.
<i>left</i>	Pointer to left substring (modified)
<i>right</i>	Pointer to right substring (modified)
<i>sc</i>	Split character.

**5.71.3.26** int *ds\_str\_strchr* ( *ds\_str str*, const char *ch*, const int *start* )

Returns index of first occurrence of a character.

## Parameters

<i>str</i>	The string.
<i>ch</i>	The character for which to search.
<i>start</i>	The index of the string at which to start looking. Set this to non-zero to begin searching from a point other than the first character of the string.

## Returns

The index of the first occurrence, or -1 if the character was not found.

**5.71.3.27** *ds\_str* *ds\_str\_substr\_left* ( *ds\_str str*, const size\_t *numchars* )

Returns a left substring.

## Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of left characters to return. If this is greater than the length of the string, the whole string is returned.

## Returns

A new string representing the substring.

**5.71.3.28** *ds\_str* *ds\_str\_substr\_right* ( *ds\_str str*, const size\_t *numchars* )

Returns a right substring.

## Parameters

<i>str</i>	The string.
<i>numchars</i>	The number of right characters to return. If this is greater than the length of the string, the whole string is returned.

## Returns

A new string representing the substring.

## 5.71.3.29 void ds\_str\_trim ( ds\_str str )

Trims leading and trailing whitespace in-place.

## Parameters

<i>str</i>	The string.
------------	-------------

## 5.71.3.30 void ds\_str\_trim\_leading ( ds\_str str )

Trims leading whitespace in-place.

## Parameters

<i>str</i>	The string.
------------	-------------

## 5.71.3.31 void ds\_str\_trim\_trailing ( ds\_str str )

Trims trailing whitespace in-place.

## Parameters

<i>str</i>	The string.
------------	-------------

## 5.71.3.32 ds\_str ds\_str\_trunc ( ds\_str str, const size\_t length )

Truncates a string.

## Parameters

<i>str</i>	The string.
<i>length</i>	The new length to which to truncate.

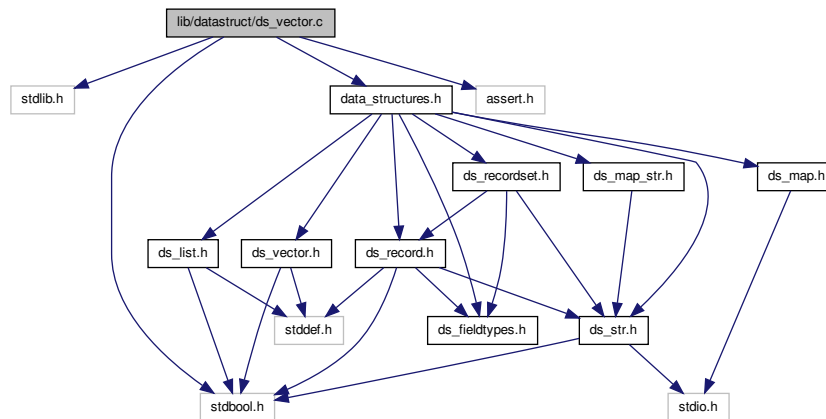
## Returns

The original string, or `NULL` on failure.

## 5.72 lib/datastruct/ds\_vector.c File Reference

Implementation of generic doubly-linked vector data structure.

```
#include <stdlib.h>
#include <stdbool.h>
#include <assert.h>
#include "data_structures.h"
Include dependency graph for ds_vector.c:
```



## Data Structures

- struct [ds\\_vector](#)

## Functions

- [ds\\_vector ds\\_vector\\_create](#) (const size\_t size, const bool free\_on\_delete, void(\*destructor)(void \*))  
*Creates a new vector.*
- void [ds\\_vector\\_destroy](#) (ds\_vector vector)  
*Destroys a vector and frees any associated resources.*
- void [ds\\_vector\\_destructor](#) (void \*vector)  
*A vector destructor function.*
- void [ds\\_vector\\_clear](#) (ds\_vector vector)  
*Clears all the elements in a vector.*
- void [ds\\_vector\\_set](#) (ds\_vector vector, const size\_t index, void \*element)  
*Sets an element of a vector.*
- void \* [ds\\_vector\\_element](#) (ds\_vector vector, const size\_t index)  
*Retrieves the data at a specified index.*
- size\_t [ds\\_vector\\_size](#) (ds\_vector vector)  
*Returns the size of a vector.*
- void [ds\\_vector\\_seek\\_start](#) (ds\_vector vector)  
*Sets the current element to the first element of a vector.*
- void \* [ds\\_vector\\_get\\_next\\_data](#) (ds\_vector vector)  
*Returns the next element of the vector.*

### 5.72.1 Detailed Description

Implementation of generic doubly-linked vector data structure.

## Author

Paul Griffiths

## Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.72.2 Function Documentation

## 5.72.2.1 void ds\_vector\_clear ( ds\_vector vector )

Clears all the elements in a vector.

If the vector was created with `free_on_delete`, the elements are `free()`d prior to being cleared (i.e. set to `NULL`).

## Parameters

<i>vector</i>	The vector.
---------------	-------------

## 5.72.2.2 ds\_vector ds\_vector\_create ( const size\_t size, const bool free\_on\_delete, void(\*)(void \*) destructor )

Creates a new vector.

## Parameters

<i>size</i>	The size of the vector.
<i>free_on_delete</i>	Set to <code>true</code> if the vector elements should be destroyed when removed from the vector, and when the vector itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the vector.
<i>destructor</i>	Pointer to a destructor function to use for destroying the vector elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

## Returns

A newly created vector, or `NULL` on failure.

## 5.72.2.3 void ds\_vector\_destroy ( ds\_vector vector )

Destroys a vector and frees any associated resources.

## Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

## 5.72.2.4 void ds\_vector\_destructor ( void \* vector )

A vector destructor function.

This function may be passed to `ds_vector_create()` when creating a vector of vectors. It calls `ds_vector_destroy()`, but the parameter of `ds_vector_destroy()` is not compatible with the function signature expected by `ds_vector_create()`, so this function provides an appropriate interface.

## Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

**5.72.2.5 void\* ds\_vector\_element ( ds\_vector vector, const size\_t index )**

Retrieves the data at a specified index.

## Parameters

<i>vector</i>	The vector from which to retrieve.
<i>index</i>	The index of the desired element.

## Returns

A pointer to the data, or `NULL` if the index is out of range.

**5.72.2.6 void\* ds\_vector\_get\_next\_data ( ds\_vector vector )**

Returns the next element of the vector.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

## Parameters

<i>vector</i>	The vector.
---------------	-------------

## Returns

A pointer to the next element, or `NULL` if the end of the vector has been reached.

**5.72.2.7 void ds\_vector\_seek\_start ( ds\_vector vector )**

Sets the current element to the first element of a vector.

## Parameters

<i>vector</i>	The vector.
---------------	-------------

**5.72.2.8 void ds\_vector\_set ( ds\_vector vector, const size\_t index, void \* element )**

Sets an element of a vector.

If the element is currently occupied, the existing element is `free()`d.

## Parameters

<i>vector</i>	The vector to which to set.
<i>index</i>	The index of the element to set.
<i>element</i>	The element to set.



5.72.2.9 `size_t ds_vector_size ( ds_vector vector )`

Returns the size of a vector.

## Parameters

<code>vector</code>	The vector.
---------------------	-------------

## Returns

The size of the vector.

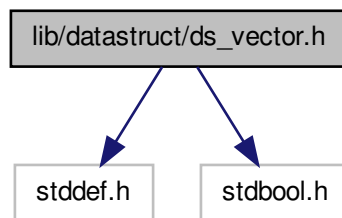
## 5.73 lib/datastruct/ds\_vector.h File Reference

Interface to generic doubly-linked vector data structure.

```
#include <stddef.h>
```

```
#include <stdbool.h>
```

Include dependency graph for ds\_vector.h:



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef struct `ds_vector` \* `ds_vector`

## Functions

- `ds_vector ds_vector_create` (const `size_t` size, const bool free\_on\_delete, void(\*destructor)(void \*))  
Creates a new vector.
- void `ds_vector_destroy` (`ds_vector` vector)  
Destroys a vector and frees any associated resources.

- void `ds_vector_destructor` (void \*vector)  
*A vector destructor function.*
- void `ds_vector_clear` (`ds_vector` vector)  
*Clears all the elements in a vector.*
- void `ds_vector_set` (`ds_vector` vector, const size\_t index, void \*element)  
*Sets an element of a vector.*
- void \* `ds_vector_element` (`ds_vector` vector, const size\_t index)  
*Retrieves the data at a specified index.*
- size\_t `ds_vector_size` (`ds_vector` vector)  
*Returns the size of a vector.*
- void `ds_vector_seek_start` (`ds_vector` vector)  
*Sets the current element to the first element of a vector.*
- void \* `ds_vector_get_next_data` (`ds_vector` vector)  
*Returns the next element of the vector.*

### 5.73.1 Detailed Description

Interface to generic doubly-linked vector data structure.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.73.2 Typedef Documentation

#### 5.73.2.1 typedef struct `ds_vector*` `ds_vector`

Typedef for opaque vector datatype

### 5.73.3 Function Documentation

#### 5.73.3.1 void `ds_vector_clear` ( `ds_vector` vector )

Clears all the elements in a vector.

If the vector was created with `free_on_delete`, the elements are `free()`d prior to being cleared (i.e. set to NULL).

#### Parameters

<i>vector</i>	The vector.
---------------	-------------

#### 5.73.3.2 `ds_vector` `ds_vector_create` ( const size\_t size, const bool *free\_on\_delete*, void(\*) (void \*) *destructor* )

Creates a new vector.

## Parameters

<i>size</i>	The size of the vector.
<i>free_on_delete</i>	Set to <code>true</code> if the vector elements should be destroyed when removed from the vector, and when the vector itself is destroyed. If set to <code>false</code> , the caller is responsible for destroying the elements prior to destroying the vector.
<i>destructor</i>	Pointer to a destructor function to use for destroying the vector elements, when <code>free_on_delete</code> is true. If this is set to <code>NULL</code> , <code>free()</code> from the standard C library will be used to destroy the elements.

## Returns

A newly created vector, or `NULL` on failure.

## 5.73.3.3 void ds\_vector\_destroy ( ds\_vector vector )

Destroys a vector and frees any associated resources.

## Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

## 5.73.3.4 void ds\_vector\_destructor ( void \* vector )

A vector destructor function.

This function may be passed to `ds_vector_create()` when creating a vector of vectors. It calls `ds_vector_destroy()`, but the parameter of `ds_vector_destroy()` is not compatible with the function signature expected by `ds_vector_create()`, so this function provides an appropriate interface.

## Parameters

<i>vector</i>	The vector to destroy.
---------------	------------------------

## 5.73.3.5 void\* ds\_vector\_element ( ds\_vector vector, const size\_t index )

Retrieves the data at a specified index.

## Parameters

<i>vector</i>	The vector from which to retrieve.
<i>index</i>	The index of the desired element.

## Returns

A pointer to the data, or `NULL` if the index is out of range.

## 5.73.3.6 void\* ds\_vector\_get\_next\_data ( ds\_vector vector )

Returns the next element of the vector.

This function returns the data of the "current element", and advances the current element pointer. Subsequent calls to this function will return successive elements.

## Parameters

<i>vector</i>	The vector.
---------------	-------------

## Returns

A pointer to the next element, or `NULL` if the end of the vector has been reached.

#### 5.73.3.7 void ds\_vector\_seek\_start ( ds\_vector vector )

Sets the current element to the first element of a vector.

## Parameters

<i>vector</i>	The vector.
---------------	-------------

#### 5.73.3.8 void ds\_vector\_set ( ds\_vector vector, const size\_t index, void \* element )

Sets an element of a vector.

If the element is currently occupied, the existing element is `free()`d.

## Parameters

<i>vector</i>	The vector to which to set.
<i>index</i>	The index of the element to set.
<i>element</i>	The element to set.

#### 5.73.3.9 size\_t ds\_vector\_size ( ds\_vector vector )

Returns the size of a vector.

## Parameters

<i>vector</i>	The vector.
---------------	-------------

## Returns

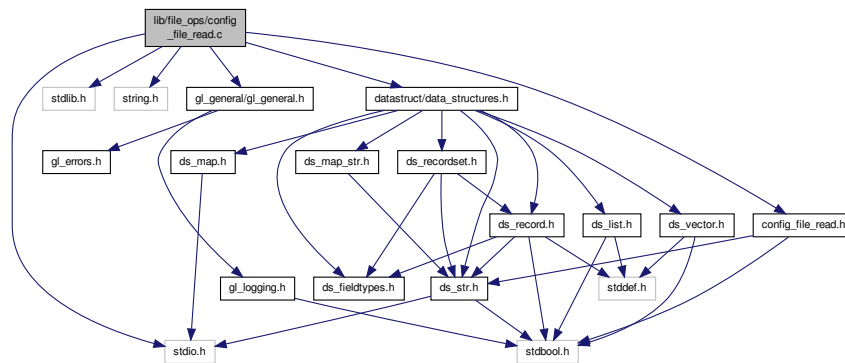
The size of the vector.

## 5.74 lib/file\_ops/config\_file\_read.c File Reference

Implementation of configuration file reading functionality.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "gl_general/gl_general.h"
#include "datastruct/data_structures.h"
#include "config_file_read.h"
```

Include dependency graph for config\_file\_read.c:



## Macros

- `#define MAX_BUFFER_SIZE 1024`
- `#define CONFIG_MAP_SIZE 100`

## Functions

- `bool config_init (void)`  
*Initializes configuration data.*
- `int config_file_read (const char *filename)`  
*Reads a configuration file and stores the key-value pairs.*
- `ds_str config_value_get (ds_str key)`  
*Returns the value associated with a key.*
- `ds_str config_value_get_cstr (const char *key)`  
*Returns the value associated with a C-style string key.*
- `void config_value_set (ds_str key, ds_str value)`  
*Sets a key-value in the configuration structure.*
- `void config_free (void)`  
*Frees the resources used by this module.*

### 5.74.1 Detailed Description

Implementation of configuration file reading functionality. This module reads configuration files in the format "key = value" and makes those values available. Leading and trailing whitespace is removed for both the key and the value. Blank lines and lines starting with a '#' are ignored in the configuration file.

#### Author

Paul Griffiths

#### Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.74.2 Macro Definition Documentation

### 5.74.2.1 `#define CONFIG_MAP_SIZE 100`

Size to use for the hash map to contain the key-value pairs

### 5.74.2.2 `#define MAX_BUFFER_SIZE 1024`

Maximum size of buffers

## 5.74.3 Function Documentation

### 5.74.3.1 `int config_file_read ( const char * filename )`

Reads a configuration file and stores the key-value pairs.

#### Parameters

<i>filename</i>	The name of the configuration file.
-----------------	-------------------------------------

#### Returns

`CONFIG_FILE_OK` on success, `CONFIG_FILE_NO_FILE` if the specified file could not be opened for reading, `CONFIG_FILE_MALFORMED_FILE` if the configuration file was improperly formed.

### 5.74.3.2 `void config_free ( void )`

Frees the resources used by this module.

The user should make copies of any required keys or values prior to calling this function. This function need not be called if `config_file_read()` returned an error.

### 5.74.3.3 `bool config_init ( void )`

Initializes configuration data.

#### Returns

`true` on success, `false` on failure.

### 5.74.3.4 `ds_str config_value_get ( ds_str key )`

Returns the value associated with a key.

#### Parameters

<i>key</i>	The specified key.
------------	--------------------

#### Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

#### 5.74.3.5 ds\_str config\_value\_get\_cstr ( const char \* key )

Returns the value associated with a C-style string key.

##### Parameters

<i>key</i>	The specified key.
------------	--------------------

##### Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

#### 5.74.3.6 void config\_value\_set ( ds\_str key, ds\_str value )

Sets a key-value in the configuration structure.

##### Parameters

<i>key</i>	The key.
<i>value</i>	The value.

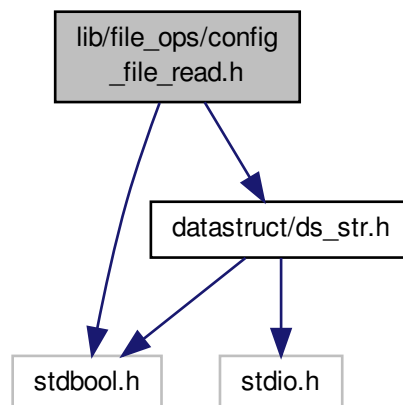
## 5.75 lib/file\_ops/config\_file\_read.h File Reference

Interface to configuration file reading functionality.

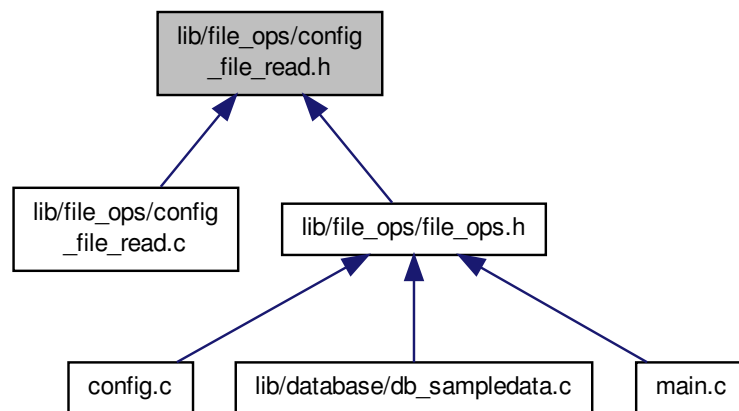
```
#include <stdbool.h>
```

```
#include "datastruct/ds_str.h"
```

Include dependency graph for config\_file\_read.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define CONFIG_FILE_OK 0`
- `#define CONFIG_FILE_NO_FILE 1`
- `#define CONFIG_FILE_MALFORMED_FILE 2`

## Functions

- `bool config_init (void)`  
*Initializes configuration data.*
- `int config_file_read (const char *filename)`  
*Reads a configuration file and stores the key-value pairs.*
- `void config_free (void)`  
*Frees the resources used by this module.*
- `ds_str config_value_get (ds_str key)`  
*Returns the value associated with a key.*
- `ds_str config_value_get_cstr (const char *key)`  
*Returns the value associated with a C-style string key.*
- `void config_value_set (ds_str key, ds_str value)`  
*Sets a key-value in the configuration structure.*

### 5.75.1 Detailed Description

Interface to configuration file reading functionality. This module reads configuration files in the format "key = value" and makes those values available. Leading and trailing whitespace is removed for both the key and the value. Blank lines and lines starting with a '#' are ignored in the configuration file.

#### Author

Paul Griffiths



## Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.75.2 Macro Definition Documentation

### 5.75.2.1 `#define CONFIG_FILE_MALFORMED_FILE 2`

Return status when configuration file is improperly formed

### 5.75.2.2 `#define CONFIG_FILE_NO_FILE 1`

Return status when unable to open file for reading

### 5.75.2.3 `#define CONFIG_FILE_OK 0`

Return status for success

## 5.75.3 Function Documentation

### 5.75.3.1 `int config_file_read ( const char * filename )`

Reads a configuration file and stores the key-value pairs.

#### Parameters

<i>filename</i>	The name of the configuration file.
-----------------	-------------------------------------

#### Returns

`CONFIG_FILE_OK` on success, `CONFIG_FILE_NO_FILE` if the specified file could not be opened for reading, `CONFIG_FILE_MALFORMED_FILE` if the configuration file was improperly formed.

### 5.75.3.2 `void config_free ( void )`

Frees the resources used by this module.

The user should make copies of any required keys or values prior to calling this function. This function need not be called if `config_file_read()` returned an error.

### 5.75.3.3 `bool config_init ( void )`

Initializes configuration data.

#### Returns

`true` on success, `false` on failure.

### 5.75.3.4 `ds_str config_value_get ( ds_str key )`

Returns the value associated with a key.

## Parameters

<i>key</i>	The specified key.
------------	--------------------

## Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

**5.75.3.5 ds\_str config\_value\_get\_cstr ( const char \* key )**

Returns the value associated with a C-style string key.

## Parameters

<i>key</i>	The specified key.
------------	--------------------

## Returns

A pointer to the associated value, or `NULL` if the key was not present in the configuration file. The caller should not modify the string to which the pointer points.

**5.75.3.6 void config\_value\_set ( ds\_str key, ds\_str value )**

Sets a key-value in the configuration structure.

## Parameters

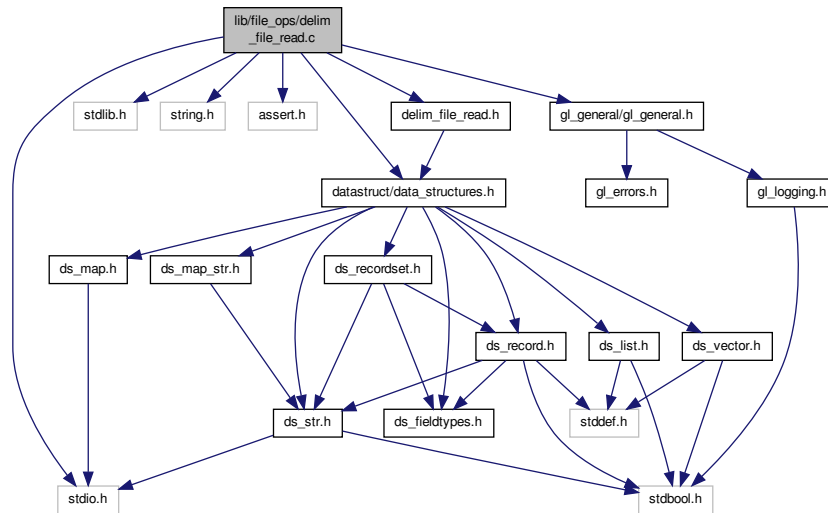
<i>key</i>	The key.
<i>value</i>	The value.

**5.76 lib/file\_ops/delim\_file\_read.c File Reference**

Implementation of delimited file reading functionality.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "gl_general/gl_general.h"
#include "datastruct/data_structures.h"
#include "delim_file_read.h"
```

Include dependency graph for `delim_file_read.c`:



## Macros

- `#define MAX_LINE_SIZE 1024`

## Functions

- `ds_recordset delim_file_read` (const char \*filename, const char delim)

Constructs a `ds_recordset` from a delimited file.

### 5.76.1 Detailed Description

Implementation of delimited file reading functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.76.2 Macro Definition Documentation

#### 5.76.2.1 `#define MAX_LINE_SIZE 1024`

Maximum size of buffers

### 5.76.3 Function Documentation

#### 5.76.3.1 `ds_recordset delim_file_read ( const char * filename, const char delim )`

Constructs a [ds\\_recordset](#) from a delimited file.

##### Parameters

<i>filename</i>	The name of the delimited file.
<i>delim</i>	The delimiting character.

##### Returns

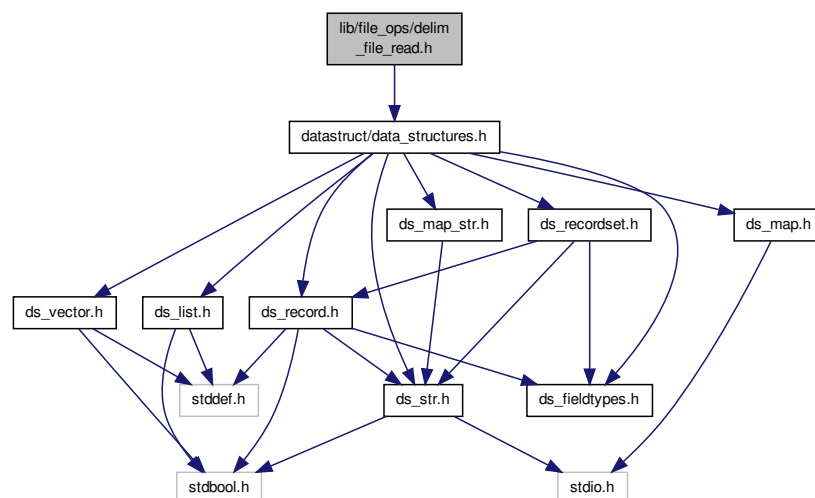
The [ds\\_recordset](#), or `NULL` on failure.

## 5.77 `lib/file_ops/delim_file_read.h` File Reference

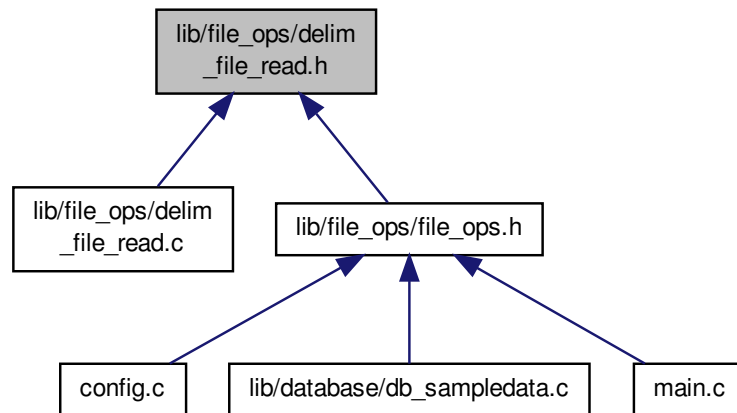
Interface to delimited file reading functionality.

```
#include "datastruct/data_structures.h"
```

Include dependency graph for `delim_file_read.h`:



This graph shows which files directly or indirectly include this file:



## Functions

- [ds\\_recordset delim\\_file\\_read](#) (const char \*filename, const char delim)  
Constructs a [ds\\_recordset](#) from a delimited file.

### 5.77.1 Detailed Description

Interface to delimited file reading functionality.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.77.2 Function Documentation

#### 5.77.2.1 ds\_recordset delim\_file\_read ( const char \* filename, const char delim )

Constructs a [ds\\_recordset](#) from a delimited file.

#### Parameters

<i>filename</i>	The name of the delimited file.
<i>delim</i>	The delimiting character.

#### Returns

The [ds\\_recordset](#), or NULL on failure.

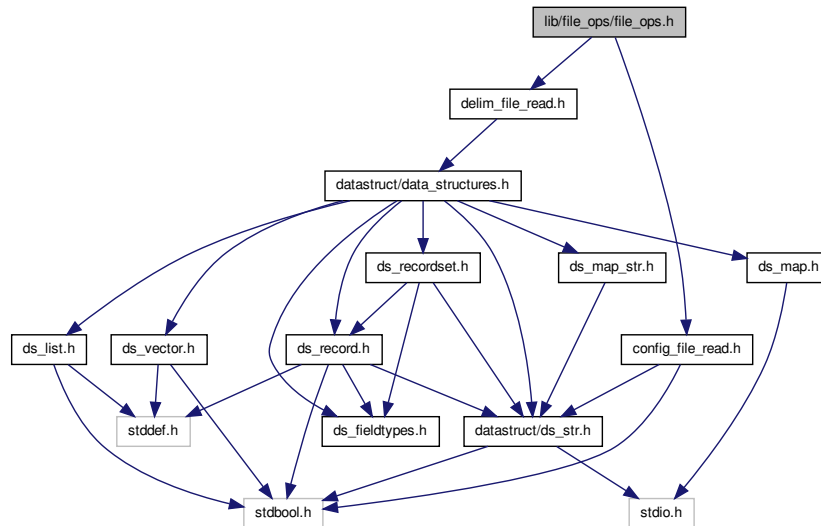
## 5.78 lib/file\_ops/file\_ops.h File Reference

User interface to file operations functionality.

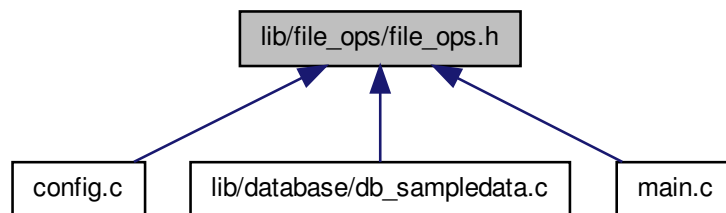
```
#include "config_file_read.h"
```

```
#include "delim_file_read.h"
```

Include dependency graph for file\_ops.h:



This graph shows which files directly or indirectly include this file:



### 5.78.1 Detailed Description

User interface to file operations functionality.

Author

Paul Griffiths

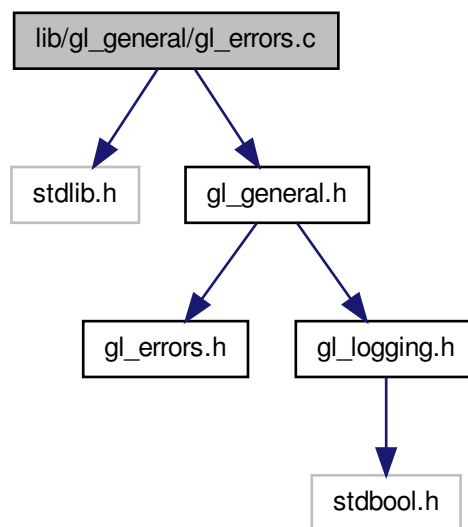
## Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.79 lib/gl\_general/gl\_errors.c File Reference

Implementation of error functionality.

```
#include <stdlib.h>
#include "gl_general.h"
Include dependency graph for gl_errors.c:
```



## Functions

- void [gl\\_error\\_quit](#) (const char \*msg)  
*Logs an error message and quits program.*

### 5.79.1 Detailed Description

Implementation of error functionality.

## Author

Paul Griffiths

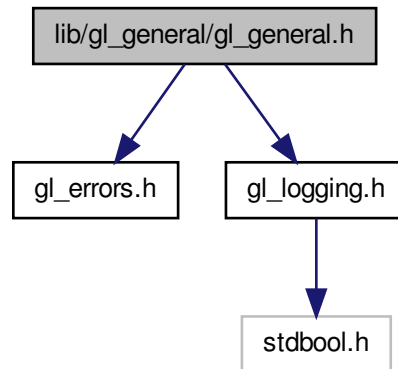
## Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

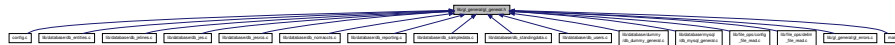




```
#include "gl_errors.h"
#include "gl_logging.h"
Include dependency graph for gl_general.h:
```



This graph shows which files directly or indirectly include this file:



### 5.81.1 Detailed Description

User interface to logging and error functionality.

**Author**

Paul Griffiths

**Copyright**

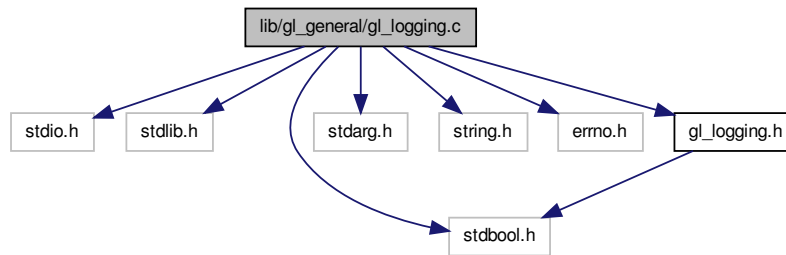
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 5.82 lib/gl\_general/gl\_logging.c File Reference

### Implementation of logging functionality.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <stdarg.h>
#include <string.h>
#include <errno.h>
#include "gl_logging.h"
```

Include dependency graph for `gl_logging.c`:



## Functions

- void `gl_set_logging` (const bool status)  
*Turns logging on or off.*
- void `gl_log_msg` (const char \*format,...)  
*Logs a message to the log file.*

### 5.82.1 Detailed Description

Implementation of logging functionality. Implementation of logging functionality. Enables debugging and other system messages to be recorded to a log file.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.82.2 Function Documentation

#### 5.82.2.1 void `gl_log_msg` ( const char \* *format*, ... )

Logs a message to the log file.

Logs a message to the log file.

#### Parameters

<i>format</i>	Format string, in same format as <code>printf()</code> .
...	Variable arguments as specified by format string.

#### 5.82.2.2 void `gl_set_logging` ( const bool *status* )

Turns logging on or off.

Turns logging on or off.

### Parameters

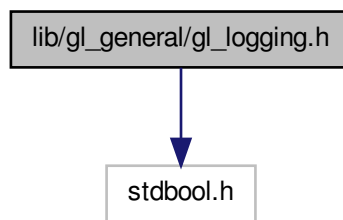
<i>status</i>	true to turn logging on, false to turn logging off.
---------------	---

### 5.83 lib/gl\_general/gl\_logging.h File Reference

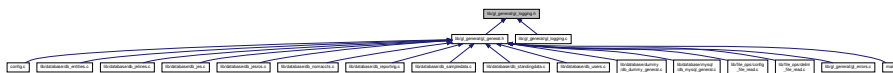
Interface to logging functionality.

```
#include <stdbool.h>
```

Include dependency graph for `gl_logging.h`:



This graph shows which files directly or indirectly include this file:



## Functions

- void **gl\_set\_logging** (const bool status)  
*Turns logging on or off.*
- void **gl\_log\_msg** (const char \*format,...)  
*Logs a message to the log file.*

### 5.83.1 Detailed Description

Interface to logging functionality. Interface to logging functionality. Enables debugging and other system messages to be recorded to a log file.

**Author**

Paul Griffiths

**Copyright**

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>



- void `print_version_message` (char \*progrname)  
*Prints a program version message.*
- void `print_help_message` (char \*progrname)  
*Prints a program help message.*
- void `test_functionality` (void)  
*Casual test function.*
- int `main` (int argc, char \*\*argv)  
*Main function.*

### 5.84.1 Detailed Description

Main function for general\_ledger. Main function for general\_ledger.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 5.84.2 Function Documentation

#### 5.84.2.1 `ds_str login ( void )`

Logs a user in and retrieves the password.

#### Returns

The password.

#### 5.84.2.2 `int main ( int argc, char ** argv )`

Main function.

Main function.

#### Returns

Exit status.

#### 5.84.2.3 `void print_help_message ( char * progrname )`

Prints a program help message.

#### Parameters

<i>progrname</i>	The program name.
------------------	-------------------

#### 5.84.2.4 void print\_usage\_message ( char \* *progrname* )

Prints a program usage message.

##### Parameters

<i>progrname</i>	The program name.
------------------	-------------------

#### 5.84.2.5 void print\_version\_message ( char \* *progrname* )

Prints a program version message.

##### Parameters

<i>progrname</i>	The program name.
------------------	-------------------

#### 5.84.2.6 void test\_functionality ( void )

Casual test function.

Used for casually testing program functionality.

# Index

- `_XOPEN_SOURCE`
  - `config.c`, 20
  - `db_dummy_general.c`, 65
- `CONFIG_FILE_OK`
  - `config_file_read.h`, 149
- `CONFIG_MAP_SIZE`
  - `config_file_read.c`, 146
- `capacity`
  - `ds_str`, 15
- `config.c`, 19
  - `_XOPEN_SOURCE`, 20
  - `get_cmdline_options`, 20
  - `get_configuration`, 20
  - `params_free`, 21
  - `params_init`, 21
- `config.h`, 21
  - `get_cmdline_options`, 22
  - `get_configuration`, 23
  - `params_free`, 23
  - `params_init`, 23
- `config_file_read`
  - `config_file_read.c`, 146
  - `config_file_read.h`, 149
- `config_file_read.c`
  - `CONFIG_MAP_SIZE`, 146
  - `config_file_read`, 146
  - `config_free`, 146
  - `config_init`, 146
  - `config_value_get`, 146
  - `config_value_get_cstr`, 146
  - `config_value_set`, 147
  - `MAX_BUFFER_SIZE`, 146
- `config_file_read.h`
  - `CONFIG_FILE_OK`, 149
  - `config_file_read`, 149
  - `config_free`, 149
  - `config_init`, 149
  - `config_value_get`, 149
  - `config_value_get_cstr`, 150
  - `config_value_set`, 150
- `config_free`
  - `config_file_read.c`, 146
  - `config_file_read.h`, 149
- `config_init`
  - `config_file_read.c`, 146
  - `config_file_read.h`, 149
- `config_value_get`
  - `config_file_read.c`, 146
  - `config_file_read.h`, 149
- `config_value_get_cstr`
  - `config_file_read.c`, 146
  - `config_file_read.h`, 150
- `config_value_set`
  - `config_file_read.c`, 147
  - `config_file_read.h`, 150
- `conn_mss`
  - `db_mysql_general.c`, 78
- `current`
  - `ds_list`, 9
  - `ds_vector`, 16
- `DS_FIELD_BOOLEAN`
  - `ds_fieldtypes.h`, 84
- `DS_FIELD_DOUBLE`
  - `ds_fieldtypes.h`, 84
- `DS_FIELD_INT`
  - `ds_fieldtypes.h`, 84
- `DS_FIELD_STRING`
  - `ds_fieldtypes.h`, 84
- `data`
  - `ds_list_element`, 11
  - `ds_str`, 15
  - `ds_vector`, 16
- `data_destructor`
  - `ds_list`, 10
  - `ds_vector`, 16
- `database`
  - `params`, 18
- `db_connect`
  - `db_connection.h`, 25
  - `db_dummy_general.c`, 65
  - `db_mysql_general.c`, 77
- `db_connection.h`
  - `db_connect`, 25
- `db_create_database_structure`
  - `db_structure.c`, 56
  - `db_structure.h`, 58
- `db_create_entities_table`
  - `db_entities.c`, 26
  - `db_entities.h`, 28
- `db_create_entities_table_sql`
  - `db_dummy_create_entities_table_sql.c`, 62
  - `db_mysql_create_entities_table_sql.c`, 67
  - `db_sql.h`, 49
- `db_create_jelines_table`
  - `db_jelines.c`, 30
  - `db_jelines.h`, 32
- `db_create_jelines_table_sql`
  - `db_mysql_create_jelines_table_sql.c`, 68

- db\_sql.h, 49
- db\_create\_jes\_table
  - db\_jes.c, 33
  - db\_jes.h, 35
- db\_create\_jes\_table\_sql
  - db\_mysql\_create\_jes\_table\_sql.c, 68
  - db\_sql.h, 49
- db\_create\_jesrcs\_table
  - db\_jesrcs.c, 36
  - db\_jesrcs.h, 38
- db\_create\_jesrcs\_table\_sql
  - db\_mysql\_create\_jesrcs\_table\_sql.c, 69
  - db\_sql.h, 49
- db\_create\_nomaccts\_table
  - db\_nomaccts.c, 39
  - db\_nomaccts.h, 41
- db\_create\_nomaccts\_table\_sql
  - db\_mysql\_create\_nomaccts\_table\_sql.c, 70
  - db\_sql.h, 49
- db\_create\_recordset\_from\_query
  - db\_dummy\_general.c, 65
  - db\_mysql\_general.c, 77
  - db\_reporting.h, 45
- db\_create\_report\_from\_query
  - db\_reporting.c, 44
  - db\_reporting.h, 45
- db\_create\_standingdata\_table
  - db\_standingdata.c, 53
  - db\_standingdata.h, 55
- db\_create\_standingdata\_table\_sql
  - db\_mysql\_create\_standingdata\_table\_sql.c, 70
  - db\_sql.h, 49
- db\_create\_users\_table
  - db\_users.c, 59
  - db\_users.h, 61
- db\_create\_users\_table\_sql
  - db\_dummy\_create\_users\_table\_sql.c, 62
  - db\_mysql\_create\_users\_table\_sql.c, 71
  - db\_sql.h, 49
- db\_current\_trial\_balance\_report
  - db\_reporting.c, 44
  - db\_reporting.h, 45
- db\_current\_trial\_balance\_report\_sql
  - db\_mysql\_current\_trial\_balance\_report\_sql.c, 71
  - db\_sql.h, 50
- db\_delete\_database\_structure
  - db\_structure.c, 56
  - db\_structure.h, 58
- db\_drop\_entities\_table
  - db\_entities.c, 26
  - db\_entities.h, 28
- db\_drop\_entities\_table\_sql
  - db\_dummy\_drop\_entities\_table\_sql.c, 63
  - db\_mysql\_drop\_entities\_table\_sql.c, 72
  - db\_sql.h, 50
- db\_drop\_jelines\_table
  - db\_jelines.c, 30
  - db\_jelines.h, 32
- db\_drop\_jelines\_table\_sql
  - db\_mysql\_drop\_jelines\_table\_sql.c, 73
  - db\_sql.h, 50
- db\_drop\_jes\_table
  - db\_jes.c, 33
  - db\_jes.h, 35
- db\_drop\_jes\_table\_sql
  - db\_mysql\_drop\_jes\_table\_sql.c, 73
  - db\_sql.h, 50
- db\_drop\_jesrcs\_table
  - db\_jesrcs.c, 36
  - db\_jesrcs.h, 38
- db\_drop\_jesrcs\_table\_sql
  - db\_mysql\_drop\_jesrcs\_table\_sql.c, 74
  - db\_sql.h, 50
- db\_drop\_nomaccts\_table
  - db\_nomaccts.c, 39
  - db\_nomaccts.h, 41
- db\_drop\_nomaccts\_table\_sql
  - db\_mysql\_drop\_nomaccts\_table\_sql.c, 74
  - db\_sql.h, 50
- db\_drop\_standingdata\_table
  - db\_standingdata.c, 53
  - db\_standingdata.h, 55
- db\_drop\_standingdata\_table\_sql
  - db\_mysql\_drop\_standingdata\_table\_sql.c, 75
  - db\_sql.h, 50
- db\_drop\_users\_table
  - db\_users.c, 59
  - db\_users.h, 61
- db\_drop\_users\_table\_sql
  - db\_dummy\_drop\_users\_table\_sql.c, 63
  - db\_mysql\_drop\_users\_table\_sql.c, 76
  - db\_sql.h, 51
- db\_dummy\_create\_entities\_table\_sql.c
  - db\_create\_entities\_table\_sql, 62
- db\_dummy\_create\_users\_table\_sql.c
  - db\_create\_users\_table\_sql, 62
- db\_dummy\_drop\_entities\_table\_sql.c
  - db\_drop\_entities\_table\_sql, 63
- db\_dummy\_drop\_users\_table\_sql.c
  - db\_drop\_users\_table\_sql, 63
- db\_dummy\_general.c
  - \_XOPEN\_SOURCE, 65
  - db\_connect, 65
  - db\_create\_recordset\_from\_query, 65
  - db\_execute\_query, 65
- db\_dummy\_list\_entities\_report\_sql.c
  - db\_list\_entities\_report\_sql, 66
- db\_dummy\_list\_users\_report\_sql.c
  - db\_list\_users\_report\_sql, 67
- db\_entities.c
  - db\_create\_entities\_table, 26
  - db\_drop\_entities\_table, 26
  - db\_list\_entities\_report, 26
- db\_entities.h
  - db\_create\_entities\_table, 28
  - db\_drop\_entities\_table, 28



- db\_list\_entities\_report, 28
- db\_execute\_query
  - db\_dummy\_general.c, 65
  - db\_mysql\_general.c, 77
  - db\_query.h, 43
- db\_jelines.c
  - db\_create\_jelines\_table, 30
  - db\_drop\_jelines\_table, 30
  - db\_list\_jelines\_report, 30
- db\_jelines.h
  - db\_create\_jelines\_table, 32
  - db\_drop\_jelines\_table, 32
  - db\_list\_jelines\_report, 32
- db\_jes.c
  - db\_create\_jes\_table, 33
  - db\_drop\_jes\_table, 33
  - db\_list\_jes\_report, 33
- db\_jes.h
  - db\_create\_jes\_table, 35
  - db\_drop\_jes\_table, 35
  - db\_list\_jes\_report, 35
- db\_jesrcs.c
  - db\_create\_jesrcs\_table, 36
  - db\_drop\_jesrcs\_table, 36
  - db\_list\_jesrcs\_report, 36
- db\_jesrcs.h
  - db\_create\_jesrcs\_table, 38
  - db\_drop\_jesrcs\_table, 38
  - db\_list\_jesrcs\_report, 38
- db\_list\_entities\_report
  - db\_entities.c, 26
  - db\_entities.h, 28
- db\_list\_entities\_report\_sql
  - db\_dummy\_list\_entities\_report\_sql.c, 66
  - db\_mysql\_list\_entities\_report\_sql.c, 78
  - db\_sql.h, 51
- db\_list\_jelines\_report
  - db\_jelines.c, 30
  - db\_jelines.h, 32
- db\_list\_jelines\_report\_sql
  - db\_mysql\_list\_jelines\_report\_sql.c, 79
  - db\_sql.h, 51
- db\_list\_jes\_report
  - db\_jes.c, 33
  - db\_jes.h, 35
- db\_list\_jes\_report\_sql
  - db\_mysql\_list\_jes\_report\_sql.c, 79
  - db\_sql.h, 51
- db\_list\_jesrcs\_report
  - db\_jesrcs.c, 36
  - db\_jesrcs.h, 38
- db\_list\_jesrcs\_report\_sql
  - db\_mysql\_list\_jesrcs\_report\_sql.c, 80
  - db\_sql.h, 51
- db\_list\_nomaccts\_report
  - db\_nomaccts.c, 39
  - db\_nomaccts.h, 41
- db\_list\_nomaccts\_report\_sql
  - db\_mysql\_list\_nomaccts\_report\_sql.c, 81
  - db\_sql.h, 51
- db\_list\_users\_report
  - db\_users.c, 59
  - db\_users.h, 61
- db\_list\_users\_report\_sql
  - db\_dummy\_list\_users\_report\_sql.c, 67
  - db\_mysql\_list\_users\_report\_sql.c, 81
  - db\_sql.h, 51
- db\_mysql\_create\_entities\_table\_sql.c
  - db\_create\_entities\_table\_sql, 67
- db\_mysql\_create\_jelines\_table\_sql.c
  - db\_create\_jelines\_table\_sql, 68
- db\_mysql\_create\_jes\_table\_sql.c
  - db\_create\_jes\_table\_sql, 68
- db\_mysql\_create\_jesrcs\_table\_sql.c
  - db\_create\_jesrcs\_table\_sql, 69
- db\_mysql\_create\_nomaccts\_table\_sql.c
  - db\_create\_nomaccts\_table\_sql, 70
- db\_mysql\_create\_standingdata\_table\_sql.c
  - db\_create\_standingdata\_table\_sql, 70
- db\_mysql\_create\_users\_table\_sql.c
  - db\_create\_users\_table\_sql, 71
- db\_mysql\_drop\_entities\_table\_sql.c
  - db\_drop\_entities\_table\_sql, 72
- db\_mysql\_drop\_jelines\_table\_sql.c
  - db\_drop\_jelines\_table\_sql, 73
- db\_mysql\_drop\_jes\_table\_sql.c
  - db\_drop\_jes\_table\_sql, 73
- db\_mysql\_drop\_jesrcs\_table\_sql.c
  - db\_drop\_jesrcs\_table\_sql, 74
- db\_mysql\_drop\_nomaccts\_table\_sql.c
  - db\_drop\_nomaccts\_table\_sql, 74
- db\_mysql\_drop\_standingdata\_table\_sql.c
  - db\_drop\_standingdata\_table\_sql, 75
- db\_mysql\_drop\_users\_table\_sql.c
  - db\_drop\_users\_table\_sql, 76
- db\_mysql\_general.c
  - conn\_mss, 78
  - db\_connect, 77
  - db\_create\_recordset\_from\_query, 77
  - db\_execute\_query, 77
  - main\_mss, 78
- db\_mysql\_list\_entities\_report\_sql.c
  - db\_list\_entities\_report\_sql, 78
- db\_mysql\_list\_jelines\_report\_sql.c
  - db\_list\_jelines\_report\_sql, 79
- db\_mysql\_list\_jes\_report\_sql.c
  - db\_list\_jes\_report\_sql, 79
- db\_mysql\_list\_jesrcs\_report\_sql.c
  - db\_list\_jesrcs\_report\_sql, 80
- db\_mysql\_list\_nomaccts\_report\_sql.c
  - db\_list\_nomaccts\_report\_sql, 81
- db\_mysql\_list\_users\_report\_sql.c
  - db\_list\_users\_report\_sql, 81
- db\_mysql\_show\_standingdata\_report\_sql.c
  - db\_show\_standingdata\_report\_sql, 82
- db\_nomaccts.c

- db\_create\_nomaccts\_table, 39
- db\_drop\_nomaccts\_table, 39
- db\_list\_nomaccts\_report, 39
- db\_nomaccts.h
  - db\_create\_nomaccts\_table, 41
  - db\_drop\_nomaccts\_table, 41
  - db\_list\_nomaccts\_report, 41
- db\_query.h
  - db\_execute\_query, 43
- db\_reporting.c
  - db\_create\_report\_from\_query, 44
  - db\_current\_trial\_balance\_report, 44
- db\_reporting.h
  - db\_create\_recordset\_from\_query, 45
  - db\_create\_report\_from\_query, 45
  - db\_current\_trial\_balance\_report, 45
- db\_show\_standingdata\_report
  - db\_standingdata.c, 53
  - db\_standingdata.h, 55
- db\_show\_standingdata\_report\_sql
  - db\_mysql\_show\_standingdata\_report\_sql.c, 82
  - db\_sql.h, 52
- db\_sql.h
  - db\_create\_entities\_table\_sql, 49
  - db\_create\_jelines\_table\_sql, 49
  - db\_create\_jes\_table\_sql, 49
  - db\_create\_jesrcs\_table\_sql, 49
  - db\_create\_nomaccts\_table\_sql, 49
  - db\_create\_standingdata\_table\_sql, 49
  - db\_create\_users\_table\_sql, 49
  - db\_current\_trial\_balance\_report\_sql, 50
  - db\_drop\_entities\_table\_sql, 50
  - db\_drop\_jelines\_table\_sql, 50
  - db\_drop\_jes\_table\_sql, 50
  - db\_drop\_jesrcs\_table\_sql, 50
  - db\_drop\_nomaccts\_table\_sql, 50
  - db\_drop\_standingdata\_table\_sql, 50
  - db\_drop\_users\_table\_sql, 51
  - db\_list\_entities\_report\_sql, 51
  - db\_list\_jelines\_report\_sql, 51
  - db\_list\_jes\_report\_sql, 51
  - db\_list\_jesrcs\_report\_sql, 51
  - db\_list\_nomaccts\_report\_sql, 51
  - db\_list\_users\_report\_sql, 51
  - db\_show\_standingdata\_report\_sql, 52
- db\_standingdata.c
  - db\_create\_standingdata\_table, 53
  - db\_drop\_standingdata\_table, 53
  - db\_show\_standingdata\_report, 53
- db\_standingdata.h
  - db\_create\_standingdata\_table, 55
  - db\_drop\_standingdata\_table, 55
  - db\_show\_standingdata\_report, 55
- db\_structure.c
  - db\_create\_database\_structure, 56
  - db\_delete\_database\_structure, 56
- db\_structure.h
  - db\_create\_database\_structure, 58
  - db\_delete\_database\_structure, 58
- db\_users.c
  - db\_create\_users\_table, 59
  - db\_drop\_users\_table, 59
  - db\_list\_users\_report, 59
- db\_users.h
  - db\_create\_users\_table, 61
  - db\_drop\_users\_table, 61
  - db\_list\_users\_report, 61
- delim\_file\_read
  - delim\_file\_read.c, 152
  - delim\_file\_read.h, 153
- delim\_file\_read.c
  - delim\_file\_read, 152
  - MAX\_LINE\_SIZE, 151
- delim\_file\_read.h
  - delim\_file\_read, 153
- ds\_fieldtypes.h
  - DS\_FIELD\_BOOLEAN, 84
  - DS\_FIELD\_DOUBLE, 84
  - DS\_FIELD\_INT, 84
  - DS\_FIELD\_STRING, 84
- ds\_field\_types
  - ds\_fieldtypes.h, 84
- ds\_fieldtypes.h
  - ds\_field\_types, 84
- ds\_list, 9
  - current, 9
  - data\_destructor, 10
  - ds\_list.h, 89
  - free\_on\_delete, 10
  - head, 10
  - length, 10
  - tail, 10
- ds\_list.c
  - ds\_list\_append, 85
  - ds\_list\_create, 85
  - ds\_list\_destroy, 86
  - ds\_list\_destructor, 86
  - ds\_list\_element, 86
  - ds\_list\_get\_next\_data, 86
  - ds\_list\_get\_prev\_data, 87
  - ds\_list\_is\_empty, 87
  - ds\_list\_length, 87
  - ds\_list\_remove\_all, 87
  - ds\_list\_remove\_tail, 88
  - ds\_list\_seek\_end, 88
  - ds\_list\_seek\_start, 88
- ds\_list.h
  - ds\_list, 89
  - ds\_list\_append, 90
  - ds\_list\_create, 90
  - ds\_list\_destroy, 90
  - ds\_list\_destructor, 90
  - ds\_list\_element, 90
  - ds\_list\_get\_next\_data, 91
  - ds\_list\_get\_prev\_data, 91
  - ds\_list\_is\_empty, 91

- [ds\\_list\\_length](#), [91](#)
  - [ds\\_list\\_remove\\_all](#), [92](#)
  - [ds\\_list\\_remove\\_tail](#), [92](#)
  - [ds\\_list\\_seek\\_end](#), [92](#)
  - [ds\\_list\\_seek\\_start](#), [92](#)
- [ds\\_list\\_append](#)
  - [ds\\_list.c](#), [85](#)
  - [ds\\_list.h](#), [90](#)
- [ds\\_list\\_create](#)
  - [ds\\_list.c](#), [85](#)
  - [ds\\_list.h](#), [90](#)
- [ds\\_list\\_destroy](#)
  - [ds\\_list.c](#), [86](#)
  - [ds\\_list.h](#), [90](#)
- [ds\\_list\\_destructor](#)
  - [ds\\_list.c](#), [86](#)
  - [ds\\_list.h](#), [90](#)
- [ds\\_list\\_element](#), [10](#)
  - [data](#), [11](#)
  - [ds\\_list.c](#), [86](#)
  - [ds\\_list.h](#), [90](#)
  - [next](#), [11](#)
  - [previous](#), [11](#)
- [ds\\_list\\_get\\_next\\_data](#)
  - [ds\\_list.c](#), [86](#)
  - [ds\\_list.h](#), [91](#)
- [ds\\_list\\_get\\_prev\\_data](#)
  - [ds\\_list.c](#), [87](#)
  - [ds\\_list.h](#), [91](#)
- [ds\\_list\\_is\\_empty](#)
  - [ds\\_list.c](#), [87](#)
  - [ds\\_list.h](#), [91](#)
- [ds\\_list\\_length](#)
  - [ds\\_list.c](#), [87](#)
  - [ds\\_list.h](#), [91](#)
- [ds\\_list\\_remove\\_all](#)
  - [ds\\_list.c](#), [87](#)
  - [ds\\_list.h](#), [92](#)
- [ds\\_list\\_remove\\_tail](#)
  - [ds\\_list.c](#), [88](#)
  - [ds\\_list.h](#), [92](#)
- [ds\\_list\\_seek\\_end](#)
  - [ds\\_list.c](#), [88](#)
  - [ds\\_list.h](#), [92](#)
- [ds\\_list\\_seek\\_start](#)
  - [ds\\_list.c](#), [88](#)
  - [ds\\_list.h](#), [92](#)
- [ds\\_map](#), [11](#)
  - [ds\\_map.h](#), [96](#)
  - [hash\\_size](#), [12](#)
  - [lists](#), [12](#)
- [ds\\_map.c](#)
  - [ds\\_map\\_destroy](#), [94](#)
  - [ds\\_map\\_get\\_value](#), [94](#)
  - [ds\\_map\\_init](#), [94](#)
  - [ds\\_map\\_insert](#), [94](#)
  - [ds\\_map\\_print\\_all](#), [95](#)
- [ds\\_map.h](#)
  - [ds\\_map](#), [96](#)
  - [ds\\_map\\_destroy](#), [96](#)
  - [ds\\_map\\_get\\_value](#), [96](#)
  - [ds\\_map\\_init](#), [96](#)
  - [ds\\_map\\_insert](#), [97](#)
  - [ds\\_map\\_print\\_all](#), [97](#)
- [ds\\_map\\_destroy](#)
  - [ds\\_map.c](#), [94](#)
  - [ds\\_map.h](#), [96](#)
- [ds\\_map\\_get\\_value](#)
  - [ds\\_map.c](#), [94](#)
  - [ds\\_map.h](#), [96](#)
- [ds\\_map\\_init](#)
  - [ds\\_map.c](#), [94](#)
  - [ds\\_map.h](#), [96](#)
- [ds\\_map\\_insert](#)
  - [ds\\_map.c](#), [94](#)
  - [ds\\_map.h](#), [97](#)
- [ds\\_map\\_print\\_all](#)
  - [ds\\_map.c](#), [95](#)
  - [ds\\_map.h](#), [97](#)
- [ds\\_map\\_str](#), [12](#)
  - [ds\\_map\\_str.h](#), [101](#)
  - [hash\\_size](#), [12](#)
  - [lists](#), [13](#)
- [ds\\_map\\_str.c](#)
  - [ds\\_map\\_str\\_destroy](#), [99](#)
  - [ds\\_map\\_str\\_get\\_value](#), [99](#)
  - [ds\\_map\\_str\\_init](#), [99](#)
  - [ds\\_map\\_str\\_insert](#), [99](#)
- [ds\\_map\\_str.h](#)
  - [ds\\_map\\_str](#), [101](#)
  - [ds\\_map\\_str\\_destroy](#), [101](#)
  - [ds\\_map\\_str\\_get\\_value](#), [101](#)
  - [ds\\_map\\_str\\_init](#), [101](#)
  - [ds\\_map\\_str\\_insert](#), [101](#)
- [ds\\_map\\_str\\_destroy](#)
  - [ds\\_map\\_str.c](#), [99](#)
  - [ds\\_map\\_str.h](#), [101](#)
- [ds\\_map\\_str\\_get\\_value](#)
  - [ds\\_map\\_str.c](#), [99](#)
  - [ds\\_map\\_str.h](#), [101](#)
- [ds\\_map\\_str\\_init](#)
  - [ds\\_map\\_str.c](#), [99](#)
  - [ds\\_map\\_str.h](#), [101](#)
- [ds\\_map\\_str\\_insert](#)
  - [ds\\_map\\_str.c](#), [99](#)
  - [ds\\_map\\_str.h](#), [101](#)
- [ds\\_record](#), [13](#)
  - [ds\\_record.h](#), [107](#)
  - [fields](#), [13](#)
- [ds\\_record.c](#)
  - [ds\\_record\\_clear](#), [103](#)
  - [ds\\_record\\_create](#), [103](#)
  - [ds\\_record\\_destroy](#), [103](#)
  - [ds\\_record\\_destructor](#), [104](#)
  - [ds\\_record\\_get\\_field](#), [104](#)
  - [ds\\_record\\_get\\_next\\_data](#), [104](#)

- ds\_record\_make\_delim\_string, 104
- ds\_record\_make\_values\_string, 104
- ds\_record\_seek\_start, 105
- ds\_record\_set\_field, 105
- ds\_record\_size, 105
- ds\_record\_tokenize, 105
- ds\_record.h
  - ds\_record, 107
  - ds\_record\_clear, 107
  - ds\_record\_create, 107
  - ds\_record\_destroy, 108
  - ds\_record\_destructor, 108
  - ds\_record\_get\_field, 108
  - ds\_record\_get\_next\_data, 108
  - ds\_record\_make\_delim\_string, 108
  - ds\_record\_make\_values\_string, 109
  - ds\_record\_seek\_start, 109
  - ds\_record\_set\_field, 109
  - ds\_record\_size, 109
  - ds\_record\_tokenize, 110
- ds\_record\_clear
  - ds\_record.c, 103
  - ds\_record.h, 107
- ds\_record\_create
  - ds\_record.c, 103
  - ds\_record.h, 107
- ds\_record\_destroy
  - ds\_record.c, 103
  - ds\_record.h, 108
- ds\_record\_destructor
  - ds\_record.c, 104
  - ds\_record.h, 108
- ds\_record\_get\_field
  - ds\_record.c, 104
  - ds\_record.h, 108
- ds\_record\_get\_next\_data
  - ds\_record.c, 104
  - ds\_record.h, 108
- ds\_record\_make\_delim\_string
  - ds\_record.c, 104
  - ds\_record.h, 108
- ds\_record\_make\_values\_string
  - ds\_record.c, 104
  - ds\_record.h, 109
- ds\_record\_seek\_start
  - ds\_record.c, 105
  - ds\_record.h, 109
- ds\_record\_set\_field
  - ds\_record.c, 105
  - ds\_record.h, 109
- ds\_record\_size
  - ds\_record.c, 105
  - ds\_record.h, 109
- ds\_record\_tokenize
  - ds\_record.c, 105
  - ds\_record.h, 110
- ds\_recordset, 14
  - ds\_recordset.h, 115
  - field\_lengths, 14
  - headers, 14
  - num\_fields, 14
  - records, 14
  - types, 15
- ds\_recordset.c
  - ds\_recordset\_add\_record, 111
  - ds\_recordset\_create, 111
  - ds\_recordset\_destroy, 112
  - ds\_recordset\_get\_next\_insert\_query, 112
  - ds\_recordset\_get\_text\_report, 112
  - ds\_recordset\_next\_record, 112
  - ds\_recordset\_num\_fields, 113
  - ds\_recordset\_num\_records, 113
  - ds\_recordset\_seek\_start, 113
  - ds\_recordset\_set\_headers, 113
  - ds\_recordset\_set\_type, 113
- ds\_recordset.h
  - ds\_recordset, 115
  - ds\_recordset\_add\_record, 115
  - ds\_recordset\_create, 116
  - ds\_recordset\_destroy, 116
  - ds\_recordset\_get\_next\_insert\_query, 116
  - ds\_recordset\_get\_text\_report, 116
  - ds\_recordset\_next\_record, 116
  - ds\_recordset\_num\_fields, 117
  - ds\_recordset\_num\_records, 117
  - ds\_recordset\_seek\_start, 117
  - ds\_recordset\_set\_headers, 117
  - ds\_recordset\_set\_type, 117
- ds\_recordset\_add\_record
  - ds\_recordset.c, 111
  - ds\_recordset.h, 115
- ds\_recordset\_create
  - ds\_recordset.c, 111
  - ds\_recordset.h, 116
- ds\_recordset\_destroy
  - ds\_recordset.c, 112
  - ds\_recordset.h, 116
- ds\_recordset\_get\_next\_insert\_query
  - ds\_recordset.c, 112
  - ds\_recordset.h, 116
- ds\_recordset\_get\_text\_report
  - ds\_recordset.c, 112
  - ds\_recordset.h, 116
- ds\_recordset\_next\_record
  - ds\_recordset.c, 112
  - ds\_recordset.h, 116
- ds\_recordset\_num\_fields
  - ds\_recordset.c, 113
  - ds\_recordset.h, 117
- ds\_recordset\_num\_records
  - ds\_recordset.c, 113
  - ds\_recordset.h, 117
- ds\_recordset\_seek\_start
  - ds\_recordset.c, 113
  - ds\_recordset.h, 117
- ds\_recordset\_set\_headers

- ds\_recordset.c, 113
  - ds\_recordset.h, 117
- ds\_recordset\_set\_type
  - ds\_recordset.c, 113
  - ds\_recordset.h, 117
- ds\_str, 15
  - capacity, 15
  - data, 15
  - ds\_str.h, 130
  - length, 15
- ds\_str.c
  - ds\_str\_assign, 120
  - ds\_str\_assign\_cstr, 120
  - ds\_str\_char\_at\_index, 120
  - ds\_str\_clear, 121
  - ds\_str\_compare, 121
  - ds\_str\_compare\_cstr, 121
  - ds\_str\_concat, 121
  - ds\_str\_concat\_cstr, 122
  - ds\_str\_create, 122
  - ds\_str\_create\_direct, 122
  - ds\_str\_create\_sprintf, 122
  - ds\_str\_cstr, 123
  - ds\_str\_decorate, 123
  - ds\_str\_destroy, 123
  - ds\_str\_destructor, 123
  - ds\_str\_doubleval, 123
  - ds\_str\_dup, 124
  - ds\_str\_getline, 124
  - ds\_str\_hash, 124
  - ds\_str\_intval, 124
  - ds\_str\_is\_alnum, 125
  - ds\_str\_is\_empty, 125
  - ds\_str\_length, 125
  - ds\_str\_size\_to\_fit, 125
  - ds\_str\_split, 126
  - ds\_str\_strchr, 126
  - ds\_str\_substr\_left, 126
  - ds\_str\_substr\_right, 126
  - ds\_str\_trim, 127
  - ds\_str\_trim\_leading, 127
  - ds\_str\_trim\_trailing, 127
  - ds\_str\_trunc, 127
- ds\_str.h
  - ds\_str, 130
  - ds\_str\_assign, 130
  - ds\_str\_assign\_cstr, 130
  - ds\_str\_char\_at\_index, 130
  - ds\_str\_clear, 131
  - ds\_str\_compare, 131
  - ds\_str\_compare\_cstr, 131
  - ds\_str\_concat, 131
  - ds\_str\_concat\_cstr, 131
  - ds\_str\_create, 132
  - ds\_str\_create\_direct, 132
  - ds\_str\_create\_sprintf, 132
  - ds\_str\_cstr, 133
  - ds\_str\_decorate, 133
  - ds\_str\_destroy, 133
  - ds\_str\_destructor, 133
  - ds\_str\_doubleval, 133
  - ds\_str\_dup, 134
  - ds\_str\_getline, 134
  - ds\_str\_hash, 134
  - ds\_str\_intval, 134
  - ds\_str\_is\_alnum, 135
  - ds\_str\_is\_empty, 135
  - ds\_str\_length, 135
  - ds\_str\_size\_to\_fit, 135
  - ds\_str\_split, 136
  - ds\_str\_strchr, 136
  - ds\_str\_substr\_left, 136
  - ds\_str\_substr\_right, 136
  - ds\_str\_trim, 137
  - ds\_str\_trim\_leading, 137
  - ds\_str\_trim\_trailing, 137
  - ds\_str\_trunc, 137
- ds\_str\_assign
  - ds\_str.c, 120
  - ds\_str.h, 130
- ds\_str\_assign\_cstr
  - ds\_str.c, 120
  - ds\_str.h, 130
- ds\_str\_char\_at\_index
  - ds\_str.c, 120
  - ds\_str.h, 130
- ds\_str\_clear
  - ds\_str.c, 121
  - ds\_str.h, 131
- ds\_str\_compare
  - ds\_str.c, 121
  - ds\_str.h, 131
- ds\_str\_compare\_cstr
  - ds\_str.c, 121
  - ds\_str.h, 131
- ds\_str\_concat
  - ds\_str.c, 121
  - ds\_str.h, 131
- ds\_str\_concat\_cstr
  - ds\_str.c, 122
  - ds\_str.h, 131
- ds\_str\_create
  - ds\_str.c, 122
  - ds\_str.h, 132
- ds\_str\_create\_direct
  - ds\_str.c, 122
  - ds\_str.h, 132
- ds\_str\_create\_sprintf
  - ds\_str.c, 122
  - ds\_str.h, 132
- ds\_str\_cstr
  - ds\_str.c, 123
  - ds\_str.h, 133
- ds\_str\_decorate
  - ds\_str.c, 123
  - ds\_str.h, 133

- ds\_str\_destroy
  - ds\_str.c, 123
  - ds\_str.h, 133
- ds\_str\_destructor
  - ds\_str.c, 123
  - ds\_str.h, 133
- ds\_str\_doubleval
  - ds\_str.c, 123
  - ds\_str.h, 133
- ds\_str\_dup
  - ds\_str.c, 124
  - ds\_str.h, 134
- ds\_str\_getline
  - ds\_str.c, 124
  - ds\_str.h, 134
- ds\_str\_hash
  - ds\_str.c, 124
  - ds\_str.h, 134
- ds\_str\_intval
  - ds\_str.c, 124
  - ds\_str.h, 134
- ds\_str\_is\_alnum
  - ds\_str.c, 125
  - ds\_str.h, 135
- ds\_str\_is\_empty
  - ds\_str.c, 125
  - ds\_str.h, 135
- ds\_str\_length
  - ds\_str.c, 125
  - ds\_str.h, 135
- ds\_str\_size\_to\_fit
  - ds\_str.c, 125
  - ds\_str.h, 135
- ds\_str\_split
  - ds\_str.c, 126
  - ds\_str.h, 136
- ds\_str\_strchr
  - ds\_str.c, 126
  - ds\_str.h, 136
- ds\_str\_substr\_left
  - ds\_str.c, 126
  - ds\_str.h, 136
- ds\_str\_substr\_right
  - ds\_str.c, 126
  - ds\_str.h, 136
- ds\_str\_trim
  - ds\_str.c, 127
  - ds\_str.h, 137
- ds\_str\_trim\_leading
  - ds\_str.c, 127
  - ds\_str.h, 137
- ds\_str\_trim\_trailing
  - ds\_str.c, 127
  - ds\_str.h, 137
- ds\_str\_trunc
  - ds\_str.c, 127
  - ds\_str.h, 137
- ds\_vector, 15
  - current, 16
  - data, 16
  - data\_destructor, 16
  - ds\_vector.h, 142
  - free\_on\_delete, 16
  - size, 16
- ds\_vector.c
  - ds\_vector\_clear, 139
  - ds\_vector\_create, 139
  - ds\_vector\_destroy, 139
  - ds\_vector\_destructor, 139
  - ds\_vector\_element, 140
  - ds\_vector\_get\_next\_data, 140
  - ds\_vector\_seek\_start, 140
  - ds\_vector\_set, 140
  - ds\_vector\_size, 140
- ds\_vector.h
  - ds\_vector, 142
  - ds\_vector\_clear, 142
  - ds\_vector\_create, 142
  - ds\_vector\_destroy, 143
  - ds\_vector\_destructor, 143
  - ds\_vector\_element, 143
  - ds\_vector\_get\_next\_data, 143
  - ds\_vector\_seek\_start, 144
  - ds\_vector\_set, 144
  - ds\_vector\_size, 144
- ds\_vector\_clear
  - ds\_vector.c, 139
  - ds\_vector.h, 142
- ds\_vector\_create
  - ds\_vector.c, 139
  - ds\_vector.h, 142
- ds\_vector\_destroy
  - ds\_vector.c, 139
  - ds\_vector.h, 143
- ds\_vector\_destructor
  - ds\_vector.c, 139
  - ds\_vector.h, 143
- ds\_vector\_element
  - ds\_vector.c, 140
  - ds\_vector.h, 143
- ds\_vector\_get\_next\_data
  - ds\_vector.c, 140
  - ds\_vector.h, 143
- ds\_vector\_seek\_start
  - ds\_vector.c, 140
  - ds\_vector.h, 144
- ds\_vector\_set
  - ds\_vector.c, 140
  - ds\_vector.h, 144
- ds\_vector\_size
  - ds\_vector.c, 140
  - ds\_vector.h, 144
- field\_lengths
  - ds\_recordset, 14
- fields
  - ds\_record, 13

- free\_on\_delete
  - ds\_list, [10](#)
  - ds\_vector, [16](#)
- get\_cmdline\_options
  - config.c, [20](#)
  - config.h, [22](#)
- get\_configuration
  - config.c, [20](#)
  - config.h, [23](#)
- gl\_error\_quit
  - gl\_errors.c, [156](#)
  - gl\_errors.h, [156](#)
- gl\_errors.c
  - gl\_error\_quit, [156](#)
- gl\_errors.h
  - gl\_error\_quit, [156](#)
- gl\_log\_msg
  - gl\_logging.c, [158](#)
  - gl\_logging.h, [160](#)
- gl\_logging.c
  - gl\_log\_msg, [158](#)
  - gl\_set\_logging, [158](#)
- gl\_logging.h
  - gl\_log\_msg, [160](#)
  - gl\_set\_logging, [160](#)
- gl\_set\_logging
  - gl\_logging.c, [158](#)
  - gl\_logging.h, [160](#)
- hash\_size
  - ds\_map, [12](#)
  - ds\_map\_str, [12](#)
- head
  - ds\_list, [10](#)
- headers
  - ds\_recordset, [14](#)
- hostname
  - params, [18](#)
- key
  - kv\_pair\_node, [17](#)
- kv\_pair\_node, [16](#)
  - key, [17](#)
  - next, [17](#)
  - value, [17](#)
- length
  - ds\_list, [10](#)
  - ds\_str, [15](#)
- lib/database/database.h, [23](#)
- lib/database/db\_connection.h, [24](#)
- lib/database/db\_entities.c, [25](#)
- lib/database/db\_entities.h, [27](#)
- lib/database/db\_internal.h, [28](#)
- lib/database/db\_jelines.c, [29](#)
- lib/database/db\_jelines.h, [31](#)
- lib/database/db\_jes.c, [32](#)
- lib/database/db\_jes.h, [34](#)
- lib/database/db\_jesrcs.c, [35](#)
- lib/database/db\_jesrcs.h, [37](#)
- lib/database/db\_nomaccts.c, [38](#)
- lib/database/db\_nomaccts.h, [40](#)
- lib/database/db\_query.h, [42](#)
- lib/database/db\_reporting.c, [43](#)
- lib/database/db\_reporting.h, [44](#)
- lib/database/db\_sampledata.c, [45](#)
- lib/database/db\_sampledata.h, [46](#)
- lib/database/db\_sql.h, [47](#)
- lib/database/db\_standingdata.c, [52](#)
- lib/database/db\_standingdata.h, [54](#)
- lib/database/db\_structure.c, [55](#)
- lib/database/db\_structure.h, [57](#)
- lib/database/db\_users.c, [58](#)
- lib/database/db\_users.h, [60](#)
- lib/database/dummy/db\_dummy\_create\_entities\_table\_sql.c, [61](#)
- lib/database/dummy/db\_dummy\_create\_users\_table\_sql.c, [62](#)
- lib/database/dummy/db\_dummy\_drop\_entities\_table\_sql.c, [62](#)
- lib/database/dummy/db\_dummy\_drop\_users\_table\_sql.c, [63](#)
- lib/database/dummy/db\_dummy\_general.c, [64](#)
- lib/database/dummy/db\_dummy\_list\_entities\_report\_sql.c, [65](#)
- lib/database/dummy/db\_dummy\_list\_users\_report\_sql.c, [66](#)
- lib/database/mysql/db\_mysql\_create\_entities\_table\_sql.c, [67](#)
- lib/database/mysql/db\_mysql\_create\_jelines\_table\_sql.c, [67](#)
- lib/database/mysql/db\_mysql\_create\_jes\_table\_sql.c, [68](#)
- lib/database/mysql/db\_mysql\_create\_jesrcs\_table\_sql.c, [69](#)
- lib/database/mysql/db\_mysql\_create\_nomaccts\_table\_sql.c, [69](#)
- lib/database/mysql/db\_mysql\_create\_standingdata\_table\_sql.c, [70](#)
- lib/database/mysql/db\_mysql\_create\_users\_table\_sql.c, [70](#)
- lib/database/mysql/db\_mysql\_current\_trial\_balance\_report\_sql.c, [71](#)
- lib/database/mysql/db\_mysql\_drop\_entities\_table\_sql.c, [72](#)
- lib/database/mysql/db\_mysql\_drop\_jelines\_table\_sql.c, [72](#)
- lib/database/mysql/db\_mysql\_drop\_jes\_table\_sql.c, [73](#)
- lib/database/mysql/db\_mysql\_drop\_jesrcs\_table\_sql.c, [73](#)
- lib/database/mysql/db\_mysql\_drop\_nomaccts\_table\_sql.c, [74](#)
- lib/database/mysql/db\_mysql\_drop\_standingdata\_table\_sql.c, [75](#)
- lib/database/mysql/db\_mysql\_drop\_users\_table\_sql.c, [75](#)

- lib/database/mysql/db\_mysql\_general.c, 76
- lib/database/mysql/db\_mysql\_list\_entities\_report\_sql.c, 78
- lib/database/mysql/db\_mysql\_list\_jelines\_report\_sql.c, 78
- lib/database/mysql/db\_mysql\_list\_jes\_report\_sql.c, 79
- lib/database/mysql/db\_mysql\_list\_jesrcs\_report\_sql.c, 80
- lib/database/mysql/db\_mysql\_list\_nomaccts\_report\_sql.c, 80
- lib/database/mysql/db\_mysql\_list\_users\_report\_sql.c, 81
- lib/database/mysql/db\_mysql\_show\_standingdata\_report\_sql.c, 81
- lib/datastruct/data\_structures.h, 82
- lib/datastruct/ds\_fieldtypes.h, 83
- lib/datastruct/ds\_list.c, 84
- lib/datastruct/ds\_list.h, 88
- lib/datastruct/ds\_map.c, 92
- lib/datastruct/ds\_map.h, 95
- lib/datastruct/ds\_map\_str.c, 97
- lib/datastruct/ds\_map\_str.h, 99
- lib/datastruct/ds\_record.c, 102
- lib/datastruct/ds\_record.h, 106
- lib/datastruct/ds\_recordset.c, 110
- lib/datastruct/ds\_recordset.h, 114
- lib/datastruct/ds\_str.c, 118
- lib/datastruct/ds\_str.h, 128
- lib/datastruct/ds\_vector.c, 137
- lib/datastruct/ds\_vector.h, 141
- lib/file\_ops/config\_file\_read.c, 144
- lib/file\_ops/config\_file\_read.h, 147
- lib/file\_ops/delim\_file\_read.c, 150
- lib/file\_ops/delim\_file\_read.h, 152
- lib/file\_ops/file\_ops.h, 154
- lib/gl\_general/gl\_errors.c, 155
- lib/gl\_general/gl\_errors.h, 156
- lib/gl\_general/gl\_general.h, 156
- lib/gl\_general/gl\_logging.c, 157
- lib/gl\_general/gl\_logging.h, 159
- lists
  - ds\_map, 12
  - ds\_map\_str, 13
- login
  - main.c, 161
- MAX\_BUFFER\_SIZE
  - config\_file\_read.c, 146
- MAX\_LINE\_SIZE
  - delim\_file\_read.c, 151
- main
  - main.c, 161
- main.c, 160
  - login, 161
  - main, 161
  - print\_help\_message, 161
  - print\_usage\_message, 161
  - print\_version\_message, 162
  - test\_functionality, 162
- main\_mss
  - db\_mysql\_general.c, 78
- next
  - ds\_list\_element, 11
  - kv\_pair\_node, 17
- num\_fields
  - ds\_recordset, 14
- params, 17
  - database, 18
  - hostname, 18
  - password, 18
  - username, 18
- params\_free
  - config.c, 21
  - config.h, 23
- params\_init
  - config.c, 21
  - config.h, 23
- password
  - params, 18
- previous
  - ds\_list\_element, 11
- print\_help\_message
  - main.c, 161
- print\_usage\_message
  - main.c, 161
- print\_version\_message
  - main.c, 162
- records
  - ds\_recordset, 14
- size
  - ds\_vector, 16
- tail
  - ds\_list, 10
- test\_functionality
  - main.c, 162
- types
  - ds\_recordset, 15
- username
  - params, 18
- value
  - kv\_pair\_node, 17